

PENSEJAJARAN SEKUENS ASAM AMINO DENGAN  
METODE SMITH-WATERMAN  
(Studi Kasus Sekuens Asam Amino Ditinjau Dari Subtitusi  
Tingkat Mutasi Asam Amino)

SKRIPSI

oleh:

MUHAMMAD BAKRI  
0410960037 – 96



PROGRAM STUDI ILMU KOMPUTER  
JURUSAN MATEMATIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS BRAWIJAYA  
MALANG  
2009

**PENSEJAJARAN SEKUENS ASAM AMINO DENGAN  
METODE SMITH-WATERMAN**  
**(Studi Kasus Sekuens Asam Amino Ditinjau Dari Subtitusi  
Tingkat Mutasi Asam Amino)**

**SKRIPSI**

Sebagai salah satu syarat untuk memperoleh gelar Sarjana  
Komputer dalam bidang Ilmu Komputer

oleh:  
**MUHAMMAD BAKRI**  
**0410960037-96**



**PROGRAM STUDI ILMU KOMPUTER**  
**JURUSAN MATEMATIKA**  
**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**  
**UNIVERSITAS BRAWIJAYA**  
**MALANG**  
**2009**

UNIVERSITAS BRAWIJAYA



## LEMBAR PENGESAHAN SKRIPSI

PENSEJAJARAN SEKUENS ASAM AMINO DENGAN  
METODE SMITH-WATERMAN

(Studi Kasus Sekuens Asam Amino Ditinjau Dari Subtitusi  
Tingkat Mutasi Asam Amino)

oleh :

MUHAMMAD BAKRI  
0410960037 – 96

Setelah dipertahankan di depan Majelis Penguji  
pada tanggal 08 September 2009  
dan dinyatakan memenuhi syarat untuk memperoleh gelar  
Sarjana Komputer dalam bidang Ilmu Komputer

Dosen Pembimbing I

Drs. Achmad Ridok, M.Kom.  
NIP. 132 090 392

Dosen Pembimbing II

Candra Dewi, S.Kom., M.Sc.  
NIP. 132 304 116

Mengetahui,  
Ketua Jurusan Matematika  
Fakultas MIPA Universitas Brawijaya

Dr. Agus Suryanto, M.Sc.  
NIP. 132 126 049

UNIVERSITAS BRAWIJAYA



## LEMBAR PERNYATAAN

Saya yang bertanda tangan di bawah ini :

Nama : Muhammad Bakri  
NIM : 0410960037-96  
Jurusan : Matematika  
Program Studi : Ilmu Komputer  
Penulis skripsi berjudul : Pensejajaran Sekuens Asam Amino dengan Metode Smith-Waterman (Studi Kasus Sekuens Asam Amino Ditinjau Dari Subtitusi Tingkat Mutasi Asam Amino)

Dengan ini menyatakan bahwa :

1. Isi dari skripsi yang saya buat adalah benar-benar karya sendiri dan tidak menjiplak karya orang lain, selain nama-nama yang termaktub di isi dan tertulis di daftar pustaka dalam skripsi ini.
2. Apabila dikemudian hari ternyata skripsi yang saya tulis terbukti hasil jiplakan, maka saya akan bersedia menanggung segala resiko yang akan saya terima.

Demikian pernyataan ini dibuat dengan segala kesadaran.

Malang, 28 September 2009

Yang menyatakan,

Muhammad Bakri

NIM. 0410960037

UNIVERSITAS BRAWIJAYA



**PENSEJAJARAN SEKUENS ASAM AMINO DENGAN  
METODE SMITH-WATERMAN**  
**(Studi Kasus Sekuens Asam Amino Ditinjau dari Subtitusi  
Tingkat Mutasi Asam Amino)**

**ABSTRAK**

Pewarisan sebuah sifat dari induk kepada keturunannya tidak pernah terjadi 100%. Hal tersebut dipengaruhi oleh faktor degenerasi gen dan mutasi. Penelitian yang dilakukan untuk mengetahui seberapa besar tingkat kesamaan/kemiripan antara dua macam individu adalah dengan melakukan pensejajaran terhadap sekuens asam amino/protein atau DNA. Smith-Waterman merupakan sebuah metode pensejajaran sekuens yang bersifat lokal, dimana pensejajaran dilakukan pada bagian tertentu pada sekuens dan tidak pada keseluruhan panjangnya. Penelitian ini bertujuan untuk mengimplementasikan metode Smith-Waterman pada sekuens asam amino yang mengikuti kaidah subtitusi tingkat mutasi asam amino. Subtitusi tingkat mutasi asam amino dipengaruhi oleh *gap* dan matriks subtitusi. Kriteria yang diperhatikan dalam pengujian metode ini adalah pengaruh perubahan nilai *gap* dan matriks subtitusi terhadap *identity score* dan *similarity score*.

Pengujian dilakukan dengan mensejajarkan dua buah sekuens dengan mengubah nilai parameter *gap* dan matriks subtitusi mulai dari nilai minimum hingga maksimum. Pengujian dilanjutkan dengan mensejajarkan sekuens dari beberapa individu yang berbeda. Berdasarkan pengujian yang dilakukan diketahui bahwa semakin besar nilai *gap* maka *identity score* dan *similarity score* akan semakin besar. Hasil penelitian menunjukkan metode Smith-Waterman dapat diimplementasikan dengan mengikuti kaidah subtitusi tingkat mutasi asam amino. *Gap penalty* menentukan besar kecilnya nilai *identity score* dan *similarity score*. Nilai *identity score* dan *similarity score* akan mencapai kondisi ekuilibrium terhadap nilai *gap open*. Kondisi tersebut disebabkan oleh jumlah *gap* pada hasil pensejajaran.

**Kata kunci :** Smith-Waterman, pensejajaran sekuens, *gap*, matriks subtitusi, asam amino

UNIVERSITAS BRAWIJAYA



# **AMINO ACIDS SEQUENCE ALIGNMENTS USING SMITH-WATERMAN METHOD**

## **(A Case Study on Amino Acids Reviewed from Substitution of Amino Acids Mutation Rate)**

### **ABSTRACT**

Inheritance of a trait from parent to the descendant never occur 100%. This factor is influenced by genes degeneration and mutation. Research studies conducted to determine how much the level of identity/similarity between two types of individuals by doing alignments to amino acid sequence/protein or DNA. Smith-Waterman is a local sequence alignments method, where the alignments done in certain parts of the sequence and not the entire lenght. This research aims to implement the Smith-Waterman method to amino acid sequences that follow the rule of substitution of amino acid mutation rate. Substitution of amino acid mutation rate is affected by gaps and substitution matrix. Criteria to be considered in testing this method is the influence of changes in the value of matrix gaps and substitution of identity score and similarity score.

Testing done by aligning two sequences with the changing value of the gap parameters and substitution matrix ranging from minimum to maximum value. Testing is continued with align sequences from several different individuals. Based on the tests is known that the greater gap value so the identity and similarity score will become greater. The result give the conclusion that the Smith-Waterman method can be implemented by following the rules of substitution of amino acid mutation rate. Gap penalty determines the size of the value of similarity score and identity score. Identity values and similarity score will reach an equilibrium condition to the value of an open gap. The condition is caused by a number of gaps in alignment results.

**Kata kunci :** Smith-Waterman, sequence alignments, gap, substitution matrix, amino acids

UNIVERSITAS BRAWIJAYA



## KATA PENGANTAR

Laporan ini disusun dalam rangka memenuhi syarat untuk memperoleh gelar Sarjana Komputer dalam bidang Ilmu Komputer. Penulis mengangkat tema pencepatan sekuens asam amino dengan metode Smith-Waterman dengan studi kasus pada substitusi tingkat mutasi asam amino. Sebuah studi bidang Bioinformatika yang banyak dibincangkan oleh para peneliti.

Penulis dalam menulis laporan ini banyak mengalami hambatan seperti kurangnya informasi yang dimiliki tentang penelitian Bioinformatika. Di samping itu, penulis juga sempat mengalami keterbatasan ide tentang bagaimana menyusun sebuah laporan. Penulis sangat bersyukur karena dapat menyelesaikan laporan dengan bantuan dari berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih kepada pihak-pihak berikut.

1. Ayahanda dan ibunda, serta saudara-saudara di Sulawesi Barat dan Sulawesi Selatan sebagai sumber motivasi penulis untuk menyelesaikan skripsi ini.
2. Drs. Achmad Ridok,M.Kom, selaku dosen pembimbing utama yang telah memberikan bantuan berupa masukan, arahan, serta tidak jarang berupa ide yang menunjang penyelesaian skripsi ini.
3. Candra Dewi,S.Kom,M.Sc, selaku dosen pembimbing kedua yang telah membantu penulis dalam menyempurnakan metode penulisan skripsi ini.
4. Nurul Hidayat,S.Pd.,M.Sc, selaku dosen pembimbing akademik yang banyak memberikan motivasi penulis dalam menentukan sikap untuk menyelesaikan penulisan skripsi ini.
5. Muhammad Imam,S.Si dan Widodo,S.Si, Magr.Sc,Phd selaku dosen Biologi yang telah memberikan banyak pengetahuan kepada penulis tentang materi bahasan Bioinformatika.
6. Rekan-rekan komunitas SIMPAI yang telah memberikan banyak wawasan kepada penulis tentang penulisan laporan skripsi.
7. Rekan-rekan di Program Studi Ilmu Komputer Universitas Brawijaya yang telah membantu penulis dalam merampungkan laporan ini.

8. Semua pihak yang telah membantu dalam penyusunan laporan ini yang tidak dapat penulis sebutkan satu per satu. Akhir kata, semoga tulisan ini bermanfaat bagi penulis khususnya dan semua pihak pada umumnya.

Malang, 25 Agustus 2009

Penulis

UNIVERSITAS BRAWIJAYA



## DAFTAR ISI

Halaman

<b>HALAMAN JUDUL .....</b>	i
<b>LEMBAR PENGESAHAN SKRIPSI .....</b>	viii
<b>LEMBAR PERNYATAAN .....</b>	v
<b>ABTRAK .....</b>	vii
<b>ABSTRACT .....</b>	ix
<b>KATA PENGANTAR .....</b>	xi
<b>DAFTAR ISI .....</b>	xiii
<b>DAFTAR GAMBAR .....</b>	xvii
<b>DAFTAR TABEL .....</b>	xix
<b>DAFTAR LAMPIRAN .....</b>	xxi

### BAB I PENDAHULUAN

1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	3
1.3 Batasan Masalah .....	3
1.4 Tujuan .....	3
1.5 Sistematika Penulisan .....	4

### BAB II TINJAUAN PUSTAKA

2.1 Kode Genetik .....	5
2.1.1 DNA (Deoxyribonucleic acid) .....	5
2.1.2 Struktur dasar DNA .....	5
2.1.3 Asam Amino .....	6
2.1.4 Penerjemahan DNA ke asam amino .....	8
2.2 Struktur Data .....	8
2.2.1 Array .....	9
2.2.2 Tabel <i>hash</i> .....	9
2.3 Algoritma Pensejajaran Sekuens .....	11
2.3.1 Konsep dasar algoritma pensejajaran sekuens .....	11
2.3.2 Pensejajaran global .....	12
2.3.3 Pensejajaran lokal .....	12
2.4 Matriks Skor/Matriks Subtitusi .....	13
2.4.1 Matriks skor DNA .....	13
2.4.2 BLOSUM .....	14
2.5 Gap .....	16

2.5.1 Pengertian dasar .....	16
2.5.2 Gap penalty .....	17
2.6 Algoritma Smith-Waterman .....	17
2.6.1 Dasar algoritma Smith-Waterman .....	17
2.6.2 Algoritma Smith-Waterman .....	18
2.6.3 Algoritma Smith-Waterman dan BLOSUM .....	20
<b>BAB III METODOLOGI DAN PERANCANGAN</b>	
3.1 Sumber Data .....	21
3.2 Struktur Data .....	22
3.2.1 Inisialisasi awal .....	22
3.2.2 Proses pensejajaran .....	22
3.2.3 Asosiasi residu .....	22
3.3 Deskripsi Sistem .....	23
3.4 Perancangan Sistem .....	23
3.4.1 Rancangan proses .....	23
3.4.2 Rancangan antarmuka pengguna .....	32
3.5 Rancangan Uji Coba .....	34
3.5.1 Uji coba perhitungan manual .....	36
3.5.2 Rancangan analisis .....	38
3.5.3 Rancangan validasi hasil .....	41
<b>BAB IV IMPLEMENTASI DAN PEMBAHASAN</b>	
4.1 Lingkungan Implementasi .....	43
4.2 Implementasi Sistem .....	43
4.2.1 Implementasi proses penerjemahan .....	43
4.2.2 Implementasi proses pensejajaran .....	45
4.3 Implementasi Antarmuka Pengguna .....	49
4.3.1 Halaman <i>Home</i> .....	49
4.3.2 Halaman <i>Result</i> .....	51
4.3.3 Halaman <i>Matrices</i> .....	52
4.4 Implementasi Uji Coba .....	53
4.4.1 Skenario pengujian .....	53
4.4.2 Evaluasi hasil .....	55
4.4.3 Validasi hasil .....	65
4.4.4 Analisis hasil .....	67

## BAB V PENUTUP

5.1 Kesimpulan .....	71
5.2 Saran .....	71
<b>DAFTAR PUSTAKA .....</b>	<b>73</b>
<b>LAMPIRAN-LAMPIRAN .....</b>	<b>75</b>



UNIVERSITAS BRAWIJAYA



## DAFTAR GAMBAR

Halaman

Gambar 2.1. Struktur molekul DNA .....	6
Gambar 2.2 Contoh penerjemahan DNA .....	8
Gambar 2.3. Array bilangan .....	9
Gambar 2.4. Implementasi tabel <i>hash</i> dari ADT <i>Dictionary</i> .....	10
Gambar 2.5. Pensejajaran tanpa <i>gap</i> .....	11
Gambar 2.6. Pensejajaran dengan <i>gap</i> pada salah satu sekuens ..	11
Gambar 2.7. Pensejajaran dengan <i>gap</i> pada semua sekuens .....	12
Gambar 2.8. Hasil pensejajaran global .....	12
Gambar 2.9. Hasil pensejajaran lokal .....	13
Gambar 2.10. Matriks skor DNA .....	14
Gambar 2.11. Matriks BLOSUM .....	16
Gambar 3.1. Bagan alur tahapan penelitian .....	21
Gambar 3.2. Format IUPAC .....	22
Gambar 3.3. Diagram alir proses sistem pensejajaran sekuens ..	24
Gambar 3.4. Diagram alir proses penerjemahan .....	25
Gambar 3.5. Diagram alir proses pensejajaran .....	27
Gambar 3.6. Diagram alir proses .....	28
Gambar 3.7. Diagram alir penelusuran balik ( <i>traceback</i> ) .....	29
Gambar 3.8. Diagram alir proses perhitungan <i>identity score</i> dan <i>similarity score</i> .....	31
Gambar 3.9. Rancangan antarmuka masukan .....	32
Gambar 3.10. Rancangan antarmuka hasil .....	33
Gambar 3.11. Rancangan antarmuka matriks skor dan pointer ..	34
Gambar 3.12. Hasil uji coba perhitungan manual .....	37
Gambar 4.1. Prosedur verifikasi data masukan .....	44
Gambar 4.2. Prosedur penerjemahan .....	44
Gambar 4.3. Inisialisasi .....	45
Gambar 4.4. Metode Smith-Waterman .....	46
Gambar 4.5. Penentuan Pointer .....	46
Gambar 4.6. Implementasi proses penelusuran .....	47
Gambar 4.7. Prosedur membuat daftar asam amino yang mirip..	48
Gambar 4.8. Perhitungan <i>identity score</i> dan <i>similarity score</i> ..	49
Gambar 4.9. Halaman <i>Home</i> .....	50
Gambar 4.10. Halaman <i>Result</i> .....	51

Gambar 4.11. Tampilan tabel skor .....	52
Gambar 4.12. Tampilan tabel pointer .....	53
Gambar 4.13. Grafik hubungan <i>gap open</i> dengan <i>identity score</i> .....	57
Gambar 4.14. Grafik hubungan <i>gap extend</i> dengan <i>identity score</i> .....	58
Gambar 4.15. Grafik hubungan <i>gap open</i> dengan <i>similarity score</i> .....	59
Gambar 4.16. Grafik hubungan <i>gap extend</i> dengan <i>similarity score</i> .....	60
Gambar 4.17. Grafik hubungan antara <i>gap open</i> dengan kemunculan gap .....	63
Gambar 4.18. Grafik hubungan antara <i>gap extend</i> dengan kemunculan gap .....	64
Gambar 4.19. Hasil pensejajaran dengan menggunakan EMBOSS .....	66
Gambar 4.20. Hasil pensejajaran dengan sistem yang dibuat .....	66
Gambar 4.21. Analisis hasil pensejajaran dengan sistem EMBOSS .....	69
Gambar 4.22. Analisis hasil pensejajaran dengan sistem yang dibuat .....	69

## DAFTAR TABEL

Halaman

Tabel 2.1 Dua puluh asam amino pembentuk protein .....	7
Tabel 3.1 Matriks hasil perhitungan .....	37
Tabel 3.2. Tabel hubungan <i>gap open</i> dengan <i>identity score</i> .....	38
Tabel 3.3. Tabel hubungan <i>gap open</i> dengan <i>similarity score</i> ....	38
Tabel 3.4. Tabel hubungan <i>gap extend</i> dengan <i>identity score</i> ....	39
Tabel 3.5. Tabel hubungan <i>gap extend</i> dengan <i>similarity score</i> .	39
Tabel 3.6. Tabel hubungan <i>gap open</i> dengan kemunculan gap pada hasil pensejajaran .....	40
Tabel 3.7. Tabel hubungan <i>gap extend</i> dengan kemunculan gap pada hasil pensejajaran .....	40
Tabel 3.8. Tabel hubungan matriks substitusi dengan <i>identity score</i> .....	41
Tabel 3.9. Tabel hubungan matriks substitusi dengan <i>similarity score</i> .....	41
Tabel 4.1. Data uji coba .....	54
Tabel 4.2. Hasil pengujian <i>gap open</i> terhadap <i>identity</i> dan <i>similarity score</i> berdasarkan matriks substitusi .....	56
Tabel 4.3. Hasil pengujian <i>gap extend</i> terhadap <i>identity</i> dan <i>similarity score</i> berdasarkan matriks substitusi .....	57
Tabel 4.4. Hasil pengujian terhadap pengaruh <i>gap open</i> pada pada kemunculan <i>gap</i> pada hasil pensejajaran .....	61
Tabel 4.5. Hasil pengujian terhadap pengaruh <i>gap extend</i> pada pada kemunculan <i>gap</i> pada hasil pensejajaran .....	62
Tabel 4.6. Tabel hubungan matriks substitusi dengan <i>identity</i> <i>score</i> .....	65
Tabel 4.7. Tabel hubungan matriks substitusi dengan <i>similarity</i> <i>score</i> .....	65
Tabel 4.8. Perbandingan hasil pensejajaran .....	66

UNIVERSITAS BRAWIJAYA



## DAFTAR LAMPIRAN

Lampiran

Halaman

Lampiran 1. Hasil perhitungan manual .....	75
Lampiran 2. Data uji coba untuk pengujian pertama dan kedua....	78
Lampiran 3. Data uji coba untuk pengujian ketiga .....	79



## BAB 1

# PENDAHULUAN

### 1.1 Latar Belakang

DNA (*Deoxyribo Nucleic Acid*) merupakan salah satu makromolekul yang mempunyai peranan penting pada jasad hidup. DNA adalah polimer asam nukleat yang tersusun secara sistematis dan merupakan pembawa informasi genetika yang diturunkan kepada jasad keturunannya. Informasi genetik disusun dalam bentuk kodon (*codon*) yang berupa tiga pasang basa nukleotida menentukan bentuk, struktur, maupun fisiologi suatu organisme. Informasi genetik tersebut merupakan bagian penting dalam pewarisan sebuah sifat suatu organisme pada keturunannya.

Pewarisan sebuah sifat pada sebagian besar organisme tidak pernah terjadi 100%. Hal ini banyak disebabkan oleh faktor degenerasi gen dan mutasi. Satu yang dapat dijamin bahwa sejumlah rangkaian DNA pada organisme selalu diwariskan pada keturunannya. Rangkaian DNA tersebut akan menentukan bahwa sebuah organisme memiliki kekerabatan terhadap organisme lainnya.

*Alignment Analysis* (Analisis Pensejajaran) merupakan metode yang dilakukan untuk mengetahui seberapa besar tingkat kekerabatan suatu organisme terhadap organisme yang lainnya. Tingkat kekerabatan organisme dilihat dari tingkat kesamaan rangkaian DNA yang dimiliki antara organisme tersebut.

Teknologi informasi sendiri memiliki peran penting dalam memberikan kemudahan kepada peneliti di bidang biologi untuk melakukan pensejajaran DNA (Cynthia Gibas dan Per Jambeck, 2003). Salah satu metode yang dapat digunakan untuk melakukan pensejajaran DNA adalah metode Smith-Waterman. Metode ini merupakan hasil penelitian yang dilakukan oleh Temple Smith dan Michael Waterman (Smith TF dan Waterman MS, 1981). Algoritma Smith-Waterman merupakan algoritma pensejajaran lokal, artinya, algoritma ini sangat berguna untuk melakukan pensejajaran pada sekuens yang tidak sama yang terletak pada satu daerah sekuens yang mirip dalam

konteks sekuens yang panjang. Pensejajaran yang dilakukan melibatkan beberapa parameter seperti *gap*, *identity score* dan *similarity score*. Dalam perkembangannya, Smith-Waterman mengalami berbagai perubahan yang disesuaikan dengan kebutuhan penelitian.

DNA berbentuk untai-ganda (double helix) yang tersusun atas empat jenis gugus karbon, yaitu Adenine (A), Tymine (T), Cytosine (C), dan Guanine (G). Hal ini dapat diartikan bahwa DNA dapat dinyatakan sebagai rangkaian karakter (string) dari kombinasi empat karakter A, C, T, dan G. Urutan basa dari suatu segmen molekul DNA itu identik dengan urutan linear dari asam-asam amino di dalam molekul protein. Asam amino dibentuk dari tiga ikatan basa DNA yang disebut kodon dan setiap asam amino memiliki keunikan tersendiri. Asam-asam amino yang memiliki sifat struktural hampir sama cenderung mempunyai kodon sekeluarga. Misalnya kodon asam aspartat sama dengan kodon asam glutamat. Jadi misalnya suatu asam amino di dalam protein karena kekeliruan digantikan oleh asam amino lain yang mempunyai sifat sama, maka protein itu akan tetap berfungsi (Suryo, 2001). Sehingga, proses pensejajaran DNA tidak hanya berdasarkan pada tingkat kesamaan setiap gugus basa A, T, C, dan G. Namun pensejajaran itu juga dapat memberikan informasi kodon yang memiliki sifat yang sama. Penelitian yang dilakukan oleh Rasmus Wernersson dan Anders Gorm Pedersen (Wernersson dan Pedersen AG, 2003) menggunakan *reverse translation* dalam melakukan pensejajaran sekuens DNA. Penelitian tersebut dilakukan dengan menerjemahkan sekuens DNA menjadi asam amino, lalu dilakukan pensejajaran. Hasil pensejajaran tersebut diterjemahkan kembali menjadi sekuens DNA (*reverse translation*). Namun hal ini masih memiliki kekurangan karena *reverse translation* memiliki kemungkinan hilangnya beberapa informasi dari kode genetik akibat penerjemahan yang tidak sesuai dengan daftar translasi asam amino.

Oleh karena itu, berdasarkan hal di atas, skripsi ini akan membahas bagaimana mengimplementasikan metode Smith-Waterman pada dua buah sekuens asam amino yang ditinjau dari subtitusi tingkat mutasi asam amino.

## 1.2 Rumusan Masalah

Berdasarkan penjelasan latar belakang di atas, pokok permasalahan yang dikemukakan dalam skripsi ini adalah sebagai berikut.

1. Bagaimana mengimplementasikan metode Smith-Waterman dengan mengikuti kaidah substitusi tingkat mutasi asam amino.
2. Bagaimana hubungan parameter *gap penalty* dan matriks substitusi mutasi asam amino terhadap besar *identity score* dan *similarity score* pada sekuens.
3. Bagaimana pengaruh *gap penalty* dan matriks substitusi mutasi asam amino terhadap hasil pensejajaran.

## 1.3 Batasan Masalah

Sumber informasi untuk mengerjakan skripsi ini sangat besar, sehingga perlu dilakukan pembatasan terhadap masalah yang akan diselesaikan. Batasan masalah untuk skripsi ini adalah sebagai berikut.

1. Pensejajaran dilakukan pada dua buah sekuens (*pairwise alignment*).
2. Sekuens DNA harus diterjemahkan menjadi asam amino.
3. Sekuen asam amino yang disejajarkan hanya menggunakan hasil penerjemahan DNA frame tunggal (*single frame*).

## 1.4 Tujuan

Pemaparan rumusan masalah di atas telah menggambarkan tujuan dari penulisan skripsi ini. Adapun tujuan penulisan skripsi ini adalah sebagai berikut.

1. Mengimplementasikan algoritma Smith-Waterman dengan mengikuti kaidah substitusi tingkat mutasi asam amino.
2. Mengetahui hubungan parameter *gap penalty* dan matriks substitusi mutasi asam amino terhadap besar *identity score* dan *similarity score* pada sekuens.
3. Mengetahui pengaruh *gap penalty* dan matriks substitusi terhadap *similarity score* dan *identity score* sekuens.

## **1.5 Sistematika Penulisan Skripsi**

### **BAB 1 PENDAHULUAN**

Bab ini berisi tentang latar belakang masalah, rumusan masalah, batasan masalah, dan tujuan penulisan skripsi

### **BAB 2 TINJAUAN PUSTAKA**

Bab ini berisi tentang teori-teori yang menjadi acuan untuk pelaksanaan penulisan skripsi, yang meliputi teori tentang kode genetik, algoritma pencejajaran sekuens, matriks skor, dan algoritma Smith-Waterman.

### **BAB 3 METODOLOGI DAN PERANCANGAN**

Bab ini berisi tentang aliran proses atau alur dari sistem yang akan dibuat beserta pembahasannya.

### **BAB 4 IMPLEMENTASI DAN PEMBAHASAN**

Bab ini membahas tentang analisa kinerja dari program yang telah dibuat baik dari antar muka maupun proses serta pembahasan prosedur dari algoritmanya.

### **BAB 5 KESIMPULAN DAN SARAN**

Bab ini berisi tentang kesimpulan dari system yang dirancang serta saran pengembangan dari keseluruhan tahapan pembuatan skripsi ini .

## BAB 2

### TINJAUAN PUSTAKA

#### 2.1 Kode Genetik

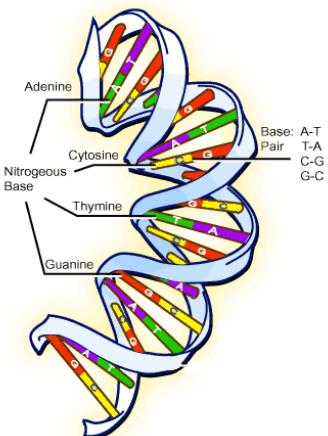
##### 2.1.1 DNA (*Deoxyribonucleic acid*)

DNA (*Deoxyribonucleic acid*) merupakan bahan genetik yang memberi informasi genetik dari sel ke sel dan satu generasi ke generasi berikutnya. Sebuah pita molekul DNA terdiri dari tiga persenyaan kimia, yaitu asam pospat, gula deoksibosa, dan basa nitrogen. Tulang punggung asam pospat-deoksiribosa selalu sama untuk berbagai segmen dari molekul DNA (Suryo, 2001).

DNA merupakan asam nukleotida yang mengandung instruksi genetik untuk pengembangan dan fungsional pada seluruh organisme hidup dan beberapa virus. Fungsi utama dari molekul DNA adalah sebagai bagian penyimpanan informasi jangka panjang. DNA juga sering diistilahkan sebagai sebuah cetak biru, resep, atau kode instruksi yang dibutuhkan untuk mengkonstruksi komponen sel lainnya seperti protein dan molekul RNA. Segmen DNA yang mengangkut informasi genetik ini disebut gen, tetapi beberapa DNA memiliki struktur tertentu (Triwibowo Yuwono, 2002).

##### 2.1.2 Struktur dasar DNA

Struktur DNA pertama kali diungkapkan oleh James Watson dan Francis Crick (1953) berdasarkan atas foto difraksi sinar X yang dibuat oleh Rosalind Franklin dan Maurice Wilkins. Berdasarkan atas data kimia dan fisik, Watson dan Crick membuat model struktur DNA yang disebut untai-ganda (*double helix*). Untai-ganda DNA tersusun oleh dua rantai polinukleotida yang terpilin. Kedua rantai mempunyai orientasi yang berlawanan (antiparalel) : rantai yang satu mempunyai orientasi  $5' \rightarrow 3'$ , sedangkan rantai yang lain berorientasi  $3' \rightarrow 5'$ . Kedua rantai tersebut berikatan karena adanya ikatan hidrogen antara basa *adenine* (A) dengan *tymine* (T), dan antara *guanine* (G) dengan *cytosine* (C), seperti yang ditunjukkan pada gambar 2.1 (Triwibowo Yuwono, 2002).



Gambar 2.1. Struktur molekul DNA

### 2.1.3 Asam amino

Asam amino adalah molekul yang mengandung amino dan gugus karboksil. Asam amino merupakan kode genetik dalam bentuk kode triplet (ditunjukkan pada tabel 2.1). Beberapa sifat dari kode triplet adalah sebagai berikut (Suryo, 2001):

1. Kode genetik ini memiliki banyak sinonim, sehingga hampir semua asam amino dinyatakan oleh lebih dari sebuah kodon. Contohnya tiga asam amino yaitu, arginin (R), serin (S), dan leusin (L) masing-masing mempunyai enam kodon sinonim. Tetapi untuk banyak kodon sinonim yang menyatakan asam amino yang sama, dua basa permulaan dari triplet adalah tetap, sedangkan basa ketiga dapat berlainan. Fleksibilitas dalam nukleotida dari suatu kodon ini dapat menolong membuat sekecil mungkin akibat adanya kesalahan.
2. Tidak ada tumpang tindih, artinya tiada satu basa tunggal pun yang dapat mengambil bagian dalam pembentukan lebih dari satu kodon sehingga 64 kodon itu semua berbeda-beda nukleotidanya.
3. Kode genetik dapat mempunyai dua arti, yaitu kodon yang sama dapat memperinci lebih dari satu asam amino. Sebagai contoh, kodon TTT biasanya merupakan kode

untuk fenilalanin (F), tetapi bila ada streptomisin dapat pula merupakan kode untuk isoleusin (I), leusin (L), dan serin (S).

4. Kode genetik tidak mempunyai tanda untuk menarik perhatian, artinya tidak ada sebuah kodon pun yang dapat diberi tambahan tanda bacaan.
5. ATG atau metionin (M) disebut juga kodon permulaan, karena kodon ini memulai sintesa rantai polipeptida.
6. TAA, TAG, TGA disebut kodon *non-sense* (tak berarti) karena tidak merupakan kode salah satu asam amino. Kodon ini juga disebut *stop codon*.
7. Kode genetik itu universal, karena kode yang sama berlaku untuk semua jenis makhluk hidup.

Tabel 2.1 Dua puluh asam amino pembentuk protein

Jenis Asam Amino	Singkatan	Simbol	Kodon
Alanine	Ala	A	GCT, GCC, GCA, GCG
Arginine	Arg	R	CGT, CGC, CGA, CGG, AGA, AGG
Asparagine	Asn	N	AAT, AAC
Aspartic acid	Asp	D	GAT, GAC
Cysteine	Cys	C	TGT, TGC
Glutamic acid	Glu	E	GAA, GAG
Glutamine	Gln	Q	CAA, CAG
Glycine	Gly	G	GGT, GGC, GGA, GGG
Histidine	His	H	CAT, CAC
Isoleucine	Ile	I	ATT, ATC, ATA
Leucine	Leu	L	CTT, CTC, CTA, CTG, TTA, TTG
Lysine	Lys	K	AAA, AAG
Methionine	Met	M	ATG
Phenylalanine	Phe	F	TTT, TTC
Proline	Pro	P	CCT, CCC, CCA, CCG
Serine	Ser	S	TCT, TCC, TCA, TCG, AGT, AGC
Threonine	Thr	T	ACT, ACC, ACA, ACG
Tryptophan	Trp	W	TGG
Tyrosine	Tyr	Y	TAT, TAC
Valine	Val	V	GTT, GTC, GTA, GTG

## 2.1.4 Penerjemahan DNA ke asam amino

Sebuah sekuens asam amino dapat diperoleh dengan melakukan proses penerjemahan DNA. Penerjemahan DNA dilakukan dengan mengikuti tabel asam amino seperti pada Tabel 2.1.

Sekuens DNA dapat diterjemahkan melalui enam kemungkinan cara. Sebuah sekuens dapat diterjemahkan secara arah mundur (*backward*) dan maju (*forward*). Setiap asam amino dalam protein di bentuk oleh tiga basa dari sekuens DNA, sehingga ada tiga kemungkinan penerjemahan pada setiap arah sebagai berikut (Cynthia Gibas and Per Jambeck, 2001) :

- 1) awal dari karakter pertama pada kodon (Frame 1)
- 2) awal dari karakter kedua pada kodon (Frame 2)
- 3) awal dari karakter ketiga pada kodon (Frame 3)

	GGA	ATA	ATA	AGA	CAC	GCC	CTG	AAG	GAG	TAC
Frame 1										
	G	I	I	R	H	A	L	K	E	Y
	GAA	TAA	TAA	GAC	ACG	CCC	TGA	AGG	AGT	AC
Frame 2										
	E	*	*	D	T	P	*	R	S	T
	AAT	AAT	AAG	ACA	CGC	CCT	GAA	GGA	GTA	C
Frame 3										
	N	N	K	T	R	P	E	G	V	

Gambar 2.2 Contoh penerjemahan DNA

Gambar 2.2 menunjukkan penerjemahan maju (*forward translation*) dari sebuah sekuens DNA. Secara natural kodon-kodon pembentuk asam amino dikelompokkan dengan spesifik. Threonine (T) selalu spesifik dengan pola A-C-X atau A-C-pyrimidine. Artinya kodon yang berawalan dengan A-C sudah dapat dipastikan merupakan asam amino threonine (T).

## 2.2 Struktur Data

Struktur data adalah cara penyimpanan atau

merepresentasikan data di dalam komputer agar dapat digunakan secara efisien. Data merupakan representasi fakta atau keterangan nyata yang disimpan, direkam, dan direpresentasikan ke dalam bentuk tulisan, gambar, suara, sinyal, atau simbol (Adi Nugroho, 2008).

Struktur data terbagi menjadi dua yaitu:

1. struktur data sederhana, misalnya array dan *record*,
2. struktur data majemuk, misalnya *stack*, *queue*, *linked list*, dan *hash table*.

### 2.2.1 Array

Array atau larik adalah sekelompok peubah tunggal atau multidimensi. Array merupakan sejumlah nilai bertipe sama yang bisa dirujuk menggunakan nama peubah/variabel yang sama diikuti indeknya.

4	13	45	11	2	32
---	----	----	----	---	----

Gambar 2.3. Array bilangan

Pada gambar 2.3, nilai-nilai pada baris yang lebih atas (4,13,45,11,2,32) adalah elemen-elemen dari array yang masing-masing dapat diakses dengan menyebutkan nama larik diikuti dengan indeknya (0,1,2,3,4,5). Gambar 2.3 menunjukkan sebuah contoh array dengan dimensi satu. Array memiliki tiga macam bentuk, yaitu (Adi Nugroho, 2008):

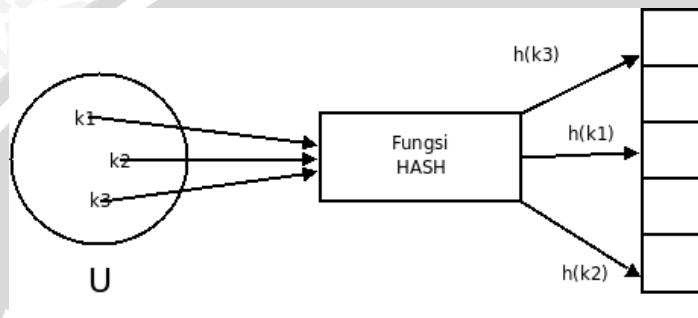
1. array dimensi satu,
2. array dimensi dua, dan
3. array multidimensi.

### 2.2.2 Tabel hash (*hash table*)

Tabel *hash* adalah struktur data yang sering digunakan untuk mengimplementasikan *Abstract Data Type* (ADT) *Dictionary*, yaitu ADT yang hanya mengizinkan pencarian, penyisipan, dan penghapusan pada elemen-elemen yang dikandung di dalamnya. Tabel *hash* tidak mendukung operasi-operasi yang akan diimplementasikan secara efisien dengan mengandalkan urutan data.

Salah satu fungsi tabel *hash* adalah untuk

mengimplementasikan tabel simbol pada *compiler*. Tabel simbol adalah suatu bentuk spesifik dari ADT *Dictionary*, yang digunakan kompiler/interpreter bahasa-bahasa pemrograman untuk memelihara informasi simbolik yang harus diakses saat menghasilkan kode yang dapat dieksekusi (*executable code*).



Gambar 2.4. Implementasi tabel *hash* dari ADT *Dictionary*

Gambar 2.4 menunjukkan bahwa  $h(k_i)$  adalah indeks (atau nilai *hash*), yang dihitung dari masing-masing nilai  $k_i \in U$ . Jika dapat dipastikan bahwa semua kunci memiliki nilai *hash* yang unik, maka operasi ADT *Dictionary* dapat diimplementasikan sebagai berikut:

```
HashTb::Search(array T, key k)
    return T[h(k)]
HashTb::Insert(array T, key k)
    return T[h(key[x])] ← x
HashTb::Delete(array T, key k)
    return T[h(k)] ← NULL
```

 (2.1)

Keunggulan pada persamaan 2.1, jika menggunakan fungsi *hash* secara benar, ukuran tabel *hash* dapat dipilih sedemikian rupa sehingga ukurannya proporsional dengan jumlah elemen yang sesungguhnya tersimpan pada tabel. Dalam hal ini, waktu yang diperlukan tentu saja sangat bergantung pada pemilihan fungsi *hash* yang sesuai (Adi Nugroho, 2008).

## 2.3 Algoritma Pensejajaran Sekuens

### 2.3.1 Konsep dasar algoritma pensejajaran sekuens

Pensejajaran sekuens merupakan metode untuk melakukan pengaturan sekuens DNA, asam amino, atau protein dengan tujuan untuk mencari daerah kesamaan secara fungsional, struktural atau hubungan evolusi antara dua buah sekuens. Sekuens yang disejajarkan direpresentasikan dalam baris matriks. Kesamaan antar sekuens akan terlihat dalam kolom matriks tersebut dengan memberikan *gap* pada sekuens agar tercipta sekuens yang sejajar (Cynthia Gibas and Per Jambeck, 2001).

Pensejajaran harus menggunakan kriteria yang tepat untuk mendapatkan hasil yang terbaik. Misalnya sekuens GCTGAACG yang ingin disejajarkan dengan CTATAATC, maka ada beberapa pilihan metode yang dapat diterapkan (Arthur Lesk, 2002).

- 1) Pensejajaran tanpa *gap*

ACGTGCTTCGTAGGATCGTT
.....: : : : : .
CGTAGGCTAGTGCTAGCTAG

Gambar 2.5. Pensejajaran tanpa *gap*

Pensejajaran tanpa *gap* dilakukan dengan mensejajarkan dua buah sekuens tanpa mempertimbangkan sekuens yang dihilangkan pada saat pensejajaran.

- 2) Pensejajaran dengan *gap* pada salah satu sekuens

ACGT - - - ACCTAGA - - - GA
: . . : . . . . . : .
AGCTCCGGTTCGTATAGCCA

Gambar 2.6. Pensejajaran dengan *gap* pada salah satu sekuens

Pensejajaran dengan *gap* pada salah satu sekuens terjadi ketika penentuan nilai *gap* diberikan pada salah satu sekuens.

- 3) Pensejajaran dengan *gap* pada semua sekuen

```
AG - - AGATC - - TACA - - G  
:: . . . . .  
AGGATC - - GGCTAGCT - - C
```

Gambar 2.7. Pensejajaran dengan *gap* pada semua sekuen

### 2.3.2 Pensejajaran global

Skenario pensejajaran yang biasanya dihadapi adalah pensejajaran dua sekuen pada seluruh panjangnya. Algoritma yang melakukan pensejajaran terhadap seluruh panjang sekuen adalah algoritma Needleman-Wunsch. Pada skenario ini, pensejajaran optimal didapatkan apabila nilai tertinggi pada pensejajaran subsekuen dengan mengikuti jejak nilai tertinggi tersebut dari pojok kiri atas sampai pojok kanan bawah. Hasil optimal hanya didapatkan melalui nilai terbaik pada matriks. Pensejajaran global akan lebih optimal ketika sekuen yang disejajarkan memiliki panjang yang sama. Hasil dari pensejajaran yang menerapkan pensejajaran global ditunjukkan pada gambar 2.8 (Cynthia Gibas and Per Jambeck, 2001).

```
FTFTALILLAVAV  
: . . . .  
F - - TAL - LLA - AV
```

Gambar 2.8. Hasil pensejajaran Global

### 2.3.3 Pensejajaran lokal

Metode pensejajaran yang sangat umum digunakan dengan mengandalkan strategi disebut pensejajaran lokal. Pensejajaran lokal merupakan pensejajaran yang mencari segmen dari dua buah sekuen yang cocok. Metode ini tidak berusaha mencari kesamaan pada seluruh sekuen, hanya pada bagian yang memiliki kesamaan yang lebih dekat berdasarkan beberapa standar (Cynthia Gibas and Per Jambeck, 2001).

Skenario yang sering dihadapi dengan pensejajaran sekuen, sering dilakukan pencarian satu sekuen dengan

membandingkan ke sekuen berbeda pada database sekuen dengan mencari sekuen baru atau mencari sekuen yang lebih panjang. Pensejajaran lokal lebih menghasilkan pensejajaran seperti yang ditunjukkan pada gambar 2.9 (Triwibowo Yuwono, 2002).

FTFTALILL-AAVV  
:::: : : :  
--FTAL-LLAAV--

Gambar 2.9. Hasil pensejajaran lokal

Jenis pemrograman dinamis yang menerapkan metode pensejajaran lokal pada dua sekuen dikenal sebagai algoritma Smith-Waterman.

## 2.4 Matriks Skor/Matriks Subtitusi

Matriks skor (*scoring matrices*) adalah tabel nilai yang menggambarkan probabilitas sebuah pasangan residu (basa atau asam amino) pada pensejajaran. Nilai dalam *scoring matrices* merupakan logaritma dari rasio dua kemungkinan. Salah satu diantara kemungkinan tersebut adalah kejadian random pada pensejajaran sekuen asam amino. Nilai ini adalah produk dari frekuensi independen dari setiap hal yang terjadi pada asam amino. Kemungkinan lainnya adalah kejadian penting pada pasangan residu dalam pensejajaran sekuen. Probabilitas ini diturunkan dari sampel aktual pada pensejajaran sekuen yang telah divalidasi. (Cynthia Gibas and Per Jambeck, 2001).

Matriks skor juga biasanya disebut dengan istilah matriks substitusi. Hal ini berdasarkan pada kemunculan gap pada saat pensejajaran akibat dari penyisipan (*insertion*) atau penghapusan (*deletion*) (Arthur Lesk, 2002).

### 2.4.1 Matriks Skor DNA

Matriks skor DNA lebih sederhana dibandingkan matriks skor pada asam amino. Hal ini dikarenakan oleh kombinasi hanya pada empat basa yaitu A,T,G, dan C. Matriks skor DNA ditunjukkan pada gambar 2.10 (Arthur Lesk, 2002).

	A	T	G	C
A	20	10	5	5
T	10	20	5	5
G	5	5	20	10
C	5	5	10	20

Gambar 2.10. Matriks skor DNA

#### 2.4.2 BLOSUM

BLOSUM (BLOcks of Amino Acid SUbstitution Matrix) merupakan matriks substitusi yang digunakan pada pencejajaran protein atau asam amino, dimana pencejajaran dilakukan pada sekuens protein yang divergen secara evolusi. BLOSUM hanya digunakan pada pencejajaran lokal (Henikoff dan Henikoff, 1992).

BLOSUM pada awalnya diperkenalkan melalui hasil pemindaian sekuens asam amino atau protein yang sangat konservatif oleh S. Henikoff dan J. Henikoff (1992). Protein atau asam amino konservatif disini berarti protein atau asam amino tersebut tidak pernah mengalami mutasi. Penelitian ini dilakukan dengan menghitung frekuensi relatif dari asam amino dan kemungkinan substitusinya. Rumus BLOSUM ditunjukkan pada persamaan 2.2 (Arthur Lesk, 2002).

$$S_{ij} = \frac{1}{\lambda} \log \left( \frac{p_{ij}}{q_i * q_j} \right) \quad (2.2)$$

dimana :

$S_{ij}$  = nilai elemen matriks

$\lambda$  = skala faktor

$p_{ij}$  = residu homolog

$q_i, q_j$  = frekuensi residu yang tidak berkorelasi

$p_{ij}$  merupakan frekuensi target dimana probabilitas terhadap pengamatan yang diharapkan bahwa i dan j saling sejajar dalam sekuens yang homolog.  $\lambda$  merupakan skala faktor untuk membulatkan semua angka skor dalam matriks menjadi bilangan bulat.  $q_i$  dan  $q_j$  merupakan residu

independen yang tidak saling berkorelasi.

Pengamatan pasangan residu selalu diharapkan sejajar dalam protein homolog dimana  $p_{ij} > q_i q_j$  dan rasio perbandingannya lebih dari satu dan positif. Residu yang memiliki korelasi positif disebut residu konservatif dan residu dengan korelasi negatif disebut residu nonkonservatif. Istilah konservatif dalam matriks substitusi BLOSUM hanya istilah statistik dan tidak berhubungan langsung dengan struktur kimia asam amino.

Contoh perhitungan matriks berikut menunjukkan bahwa pasangan *tryptophan* (W/W) memiliki nilai +11 dan *leucine* (L/L) memiliki nilai +4. Setiap pasangan residu memiliki nilai yang berbeda karena faktor nilai kedekatan antara pasangan residu. Pasangan *leucine* memiliki frekuensi korelasi yang lebih umum dari pasangan *tryptophan* yaitu  $p_{LL} = 0,0371$  dan  $p_{WW} = 0,0065$ . Frekuensi *leucine* adalah  $q_L = 0,099$  dan *tryptophan* adalah  $q_W = 0,013$ . Jika dihitung berdasarkan rumus dengan skala faktor=0,347, maka  $S_{LL} = +3,8$  dan  $S_{WW} = +10,5$ . Hasil tersebut dibulatkan menjadi  $S_{LL} = +4$  dan  $S_{WW} = +11$  (Sean R. Eddy, 2004). Matriks BLOSUM ditunjukkan pada gambar 2.11

#	Matrix made by matblas from blosum62.iiij
#	* column uses minimum score
#	BLOSUM Clustered Scoring Matrix in 1/2 Bit Units
#	Blocks Database = /data/blocks_5.0/blocks.dat
#	Cluster Percentage: >= 62
#	Entropy = 0.6979, Expected = -0.5209
A	R N D C Q E G H I L K M F P S T W Y V B Z X *
A	4 -1 -2 -2 -1 -1 0 -2 -1 -1 -1 -2 -1 0 1 -3 -2 0 -2 -1 0 -4
R	-1 5 0 -2 -3 1 0 -2 0 -3 -2 2 -1 -3 -2 -1 -1 -3 -2 -3 -1 0 -1 -4
N	-2 0 6 1 -3 0 0 0 1 -3 -3 0 -2 -3 -2 1 0 -4 -2 -3 3 0 -1 -4
D	-2 -2 1 6 -3 0 2 -1 -1 -3 4 -1 -3 -3 -1 0 -1 -4 -3 -3 4 1 -1 -4
C	0 -3 -3 -3 9 -3 -4 -3 -3 -1 -1 -3 -1 -2 -3 -1 -1 -2 -2 -1 -3 -3 -2 -4
Q	-1 1 0 0 -3 5 2 -2 0 -3 -2 1 0 -3 -1 0 -1 -2 -1 -2 0 3 -1 -4
E	-1 0 0 2 -4 2 5 -2 0 -3 -3 1 -2 -3 -1 0 -1 -3 -2 -2 1 4 -1 -4
G	0 -2 0 -1 -3 -2 -2 6 -2 -4 -4 -2 -3 -3 -2 0 -2 -2 -3 -3 -1 -2 -1 -4
H	-2 0 1 -1 -3 0 0 -2 -8 -3 -3 -1 -2 -1 -2 -1 -2 -2 2 -3 0 0 -1 -4
I	-1 -3 -3 -3 -1 -3 -3 -4 -3 4 2 -3 1 0 -3 -2 -1 -3 -1 3 -3 -3 -1 -4
L	-1 -2 -3 -4 -1 -2 -3 -4 -3 2 4 -2 2 0 -3 -2 -1 -2 -1 1 -4 -3 -1 -4
K	-1 2 0 -1 -3 1 1 -2 -1 -3 -2 5 -1 -3 -1 0 -1 -3 -2 -2 0 1 -1 -4
M	-1 -1 -2 -3 -1 0 -2 -3 -2 1 2 -1 5 0 -2 -1 -1 -1 -1 1 -3 -1 -1 -4
F	-2 -3 -3 -3 -2 -3 -3 -3 -1 0 0 -3 0 6 -4 -2 -2 1 3 -1 -3 -3 -1 -4
P	-1 -2 -2 -1 -3 -1 -1 -2 -2 -3 -3 -1 -2 -4 7 -1 -1 -4 -3 -2 -2 -1 -2 -4
S	1 -1 1 0 -1 0 0 0 -1 -2 0 0 -1 -2 -1 4 1 -3 -2 -2 0 0 0 -4
T	0 -1 0 -1 -1 -1 -1 -2 -2 -1 -1 -1 -2 -1 1 5 -2 -2 0 -1 -1 0 -4
W	-3 -3 -4 -4 -4 -2 -2 -3 -2 -2 -3 -2 -3 -1 1 -4 -3 -2 11 2 -3 -4 -3 -2 -4
Y	-2 -2 -2 -3 -3 -2 -1 -2 -3 -2 -1 -1 -2 -1 3 -3 -2 -2 2 7 -1 -3 -2 -1 -4
V	0 -3 -3 -3 -1 -2 -2 -3 -3 -3 1 -2 1 -1 -2 -2 0 -3 -1 4 -3 -2 -1 -4
B	-2 -1 1 3 4 -3 0 1 -1 0 -3 -4 0 -3 -3 -2 0 -1 -4 -3 -3 4 1 -1 -4
Z	-1 0 0 1 -3 3 4 -2 0 -3 -3 1 -1 -3 -1 0 -1 -3 -2 -2 1 4 -1 -4
X	0 -1 -1 -1 -2 -1 -1 -1 -1 -1 -1 -1 -2 0 0 -2 -1 -1 -1 -1 -1 -1 -4
*	-4 -1

(a) BLOSUM62

#	Matrix made by matblas from blosum80.iij
#	* column uses minimum score
#	BLOSUM Clustered Scoring Matrix in 1/2 Bit Units
#	Blocks Database = /data(blocks_5.0/blocks.dat
#	Cluster Percentage: >= 80
#	Entropy = 0.9868, Expected = -0.7442
A	R N D C Q E G H I L K M F P S T W Y V B Z X *
A	5 -2 -2 -2 -1 -1 -1 0 -2 -2 -2 -1 -1 -3 -1 1 0 -3 -2 0 -2 -1 -1 -6
R	-2 6 -1 -2 -4 1 -1 -3 0 -3 -3 2 -2 -4 -2 -1 -1 -4 -3 -3 -2 0 -1 -6
N	-2 -1 6 1 -3 0 -1 -1 0 -4 -4 0 -3 -4 -3 0 0 -4 -3 -4 4 0 -1 -6
D	-2 -2 1 6 -4 -1 1 -2 -2 -4 -5 -1 -4 -4 -2 -1 -1 -6 -4 -4 4 1 -1 -6
C	-1 -4 -3 -4 9 -4 -5 -4 -4 -2 -2 -4 -2 -3 -4 -2 -1 -3 -3 -1 -4 -4 -1 -6
Q	-1 1 0 -1 -4 6 2 -2 1 -3 -3 1 0 -4 -2 0 -1 -3 -2 -3 0 3 -1 -6
E	-1 -1 -1 1 -5 -2 6 -3 0 -4 -4 1 -2 -4 -2 0 -1 -4 -3 -3 1 4 -1 -6
G	0 -3 -1 -2 -4 -2 -3 6 -3 -5 -4 -2 -4 -4 -3 -1 -2 -4 -4 -4 -1 -3 -1 -6
H	-2 0 0 -2 -4 1 0 -3 8 -4 -3 -1 -2 -2 -3 -1 -2 -3 2 -4 -1 0 -1 -6
I	-2 -3 -4 -4 -2 -3 -4 -5 -4 5 1 -3 1 -1 -4 -3 -1 -3 -2 3 -4 -4 -1 -6
L	-2 -3 -4 -5 -2 -3 -4 -4 -3 1 4 -3 2 0 -3 -3 -2 -2 -2 1 -4 -3 -1 -6
K	-1 2 0 -1 -4 1 1 -2 -1 -3 -3 5 -2 -4 -1 -1 -1 -4 -3 -3 1 1 -1 -6
M	-1 -2 -3 -4 -2 0 -2 -4 -2 1 2 -2 6 0 -3 -2 -1 -2 -2 1 -3 -2 -1 -6
F	-3 -4 -4 -4 -3 -4 -4 -4 -2 -1 0 -4 0 6 -4 -3 -2 0 3 -1 -4 -4 -1 -6
P	-1 -2 -3 -2 -4 -2 -3 -3 -4 -3 -1 -3 -4 8 -1 -2 -5 -4 -3 -2 -2 -1 -6
S	1 -1 0 -1 -2 0 0 -1 -1 -3 -3 -1 -2 -3 -1 5 1 -4 -2 -2 0 0 -1 -6
T	0 -1 0 -1 -1 -1 -1 -2 -2 -1 -2 -1 -2 -2 1 5 -4 -2 0 -1 -1 -1 -6
W	-3 -4 -4 -6 -3 -3 -4 -4 -3 -3 -2 -4 -2 0 -5 -4 -4 11 2 -3 -5 -4 -1 -6
Y	-2 -3 -3 -4 -3 -2 -3 -4 2 -2 -2 -3 -2 3 -4 -2 -2 2 7 -2 -3 -3 -1 -6
V	0 -3 -4 -4 -1 -3 -3 -4 -4 3 1 -3 1 -1 -3 -2 0 -3 -2 4 -4 -3 -1 -6
B	-2 -2 4 4 -4 -0 1 -1 -1 -4 -4 -1 -3 -4 -2 0 -1 -5 -3 -4 4 0 -1 -6
Z	-1 0 0 1 -4 3 4 -3 0 -4 -3 1 -2 -4 -2 0 -1 -4 -3 -3 0 4 -1 -6
X	-1 -6
*	-6 -1

(b) BLOSUM80

Gambar 2.11. Matriks BLOSUM

Pasangan residu yang memiliki korelasi terdekat akan menjadi satu kelompok residu dan dilakukan *cluster*. Hal ini untuk mengurangi kontribusi berlebihan pada frekuensi residu. Setiap *cluster* diberi nilai persentase sesuai dengan entropi relatif *cluster*. Semakin besar nilai persentase cluster maka kelompok pasangan residu yang konservatif semakin besar. Artinya, BLOSUM80 memiliki persentase cluster 80% dengan target frekuensi yang dapat dikatakan konservatif (Henikoff dan Henikoff, 1992).

## 2.5 Gap

### 2.5.1 Pengertian dasar

*Gap* merupakan bagian yang dihilangkan atau dilewati. *Gap* yang ditemukan saat membandingkan dua sekuen ditafsirkan sebagai bagian yang pada proses evolusi mengalami mutasi, sehingga hilang (*deletion*) atau digantikan (*insertion*) oleh sekuen baru. *Gap* muncul dikarenakan oleh

beberapa hal sebagai berikut (Cynthia Gibas and Per Jambeck, 2001).

1. Mutasi tunggal, hal ini sangat umum terjadi.
2. Persilangan tidak seimbang dalam meiosis dapat menyebabkan terjadinya *insertion* dan *deletion* pada sekuens.
3. Kemungkinan kecil DNA pada saat proses replikasi mengalami perulangan dalam sekuens.
4. Penyisipan retrovirus.
5. Translokasi antara DNA dengan kromosom.

### 2.5.2 *Gap penalty*

*Gap* terbagi menjadi dua yaitu *gap open penalty* dan *gap extend penalty*. *Gap open penalty* adalah skor yang diperoleh dari inisialisasi awal dari *gap* pada sekuens. *Gap extend penalty* ditambahkan pada *gap penalty* untuk setiap *gap* pada residu. *Gap extend* memiliki nilai yang lebih kecil dari *gap open*. Nilai *gap open* merupakan bilangan bulat positif dan nilai *gap extend* merupakan bilangan desimal positif. ([www.ebi.ac.uk](http://www.ebi.ac.uk)).

## 2.6 Algoritma Smith-Waterman

### 2.6.1 Dasar algoritma Smith-Waterman

Identifikasi subsekuens homolog secara maksimal pada sebuah sekuens yang sangat panjang merupakan permasalahan penting dalam analisis sekuens molekuler. Permasalahan secara langsung hanya jika salah satu perhatian khusus pada subsekuens yang kontinu (segmen) tidak memiliki penyisipan (*insertion*) atau penghapusan (*deletion*). Permasalahan umum yang telah ditemukan solusinya dalam sebuah ekstensi pada matriks sekuens yang dikembangkan untuk mengukur nilai minimal sebuah “perlakuan”, dibutuhkan untuk mengkonversi sebuah sekuens menjadi sekuens yang lain.

Pengembangan ini pada analisis sekuens modern dimulai dengan algoritma heuristik homolog oleh Needleman-Wunsch (1970) yang pertama kali memperkenalkan metode perhitungan matriks iteratif. Algoritma dengan ketepatan

matematis juga disarankan oleh Sankoff (1972), Reichert (1973) dan Beyer (1979), tetapi metode ini tidak dapat diinterpretasikan secara biologi. Sellers (1974) dengan sukses mengembangkan perhitungan matriks dari jarak antarsekuens. Metrik (metode perhitungan) ini dibakukan oleh Waterman, dkk (1976) untuk menggabungkan *deletion/insertion* dari sebuah jarak terpisah. Metrik ini merepresentasikan nilai minimum dari “peristiwa mutasi” yang dibutuhkan untuk mengkonversi satu sekuens menjadi sekuens yang lain.

Salah satu hal menarik bahwa Smith dkk (1980) telah menunjukkan bahwa dalam beberapa kondisi, metrik umum Sellers ekuivalen dengan algoritma homolog Needleman-Wunsch (1970). Ide di atas menjadi sumber inspirasi Dr. Smith dan Dr. Waterman melakukan pengembangan untuk mencari pasangan dari segmen, satu dari dua sekuens panjang, dimana tidak ada pasangan segmen yang memiliki *identity score* tinggi (homolog). Pengukuran kesamaan yang digunakan memungkinkan adanya pemisahan panjang *deletion* dan *insertion* (Smith TF dan Waterman MS, 1989).

## 2.6.2 Algoritma Smith-Waterman

Algoritma Smith-Waterman dapat dijelaskan melalui ilustrasi berikut ini. Misalnya dua buah sekuens dinyatakan sebagai  $A = a_1, a_2, \dots, a_n$ , dan  $B = b_1, b_2, \dots, b_m$ , dimana A memiliki panjang n dan B memiliki panjang m. Kesamaan antara A dan B dinyatakan dalam  $W(A, B)$ . Untuk mencari pasangan segmen dengan derajat kesamaan yang tinggi maka dibuat matriks H dengan ordo  $(n+1) \times (m+1)$ . Penambahan nilai satu pada panjang masing-masing sekuens dimaksudkan untuk memberikan nilai awal dari matriks H. Sehingga inisialisasi awalnya ditunjukkan pada persamaan 2.3.

$$H_{i0} = H_{0j} = 0 \text{ untuk } 0 \leq i \leq n \text{ dan } 0 \leq j \leq m \quad (2.3)$$

Nilai awal dari H memiliki interpretasi bahwa  $H_{ij}$  merupakan *identity score* maksimum dari dua segmen terakhir pada  $a_i$  dan  $b_j$ . Nilai ini diperoleh relasi seperti pada persamaan 2.4.

$$H_{ij} = \max\{H_{i-1,j-1} + W(a_i, b_j); H_{i-1,j} - W(a_i, -); H_{i,j-1} - W(-, b_j)\}$$

dimana  $1 \leq i \leq n$  dan  $1 \leq j \leq m$  (2.4)

Formula untuk  $H_{ij}$  berdasarkan pada kemungkinan terakhir dari tiap segmen  $a_i$  dan  $b_j$ . Jika  $a_i$  dan  $b_j$  berasosiasi maka kesamaannya ditunjukkan pada persamaan 2.5.

$$H_{i-1,j-1} + W(a_i, b_j) \quad (2.5)$$

Asosiasi kedua residu dinyatakan dengan *match* dan *dismatch*. Nilai *match* bernilai positif dan *dismatch* bernilai negatif.  $W(a_i, -)$  dan  $W(-, b_j)$  merupakan nilai segmen terakhir dari  $a_i$  dan  $b_j$  pada panjang masing-masing residu yang akan menentukan *insertion/deletion*.

Pasangan segmen dengan tingkat kesamaan tinggi didapatkan melalui nilai maksimum pada elemen H. Elemen matriks lain yang memiliki nilai maksimum akan didapatkan secara sekuensial melalui prosedur *traceback* yang berakhir pada nilai  $H = 0$ . Pasangan segmen berikutnya dengan kesamaan maksimum didapatkan melalui prosedur *traceback* yang tidak berasosiasi dengan elemen pertama pada matriks H (Smith TF dan Waterman MS, 1989).

Algoritma Smith-Waterman dapat digambarkan dalam bentuk pseudocode sebagai berikut. (Andrzej Polanski, 1992)

```
1. best = 0
2. for j = 1 to N do
3.   H(0,j) = 0
4.   for i = 1 to M do
5.     {   H(i,0) = 0
6.       for j = 1 to N do
7.         H(i,j) = max (H(i-1, j-1) + W(ai, bj),
8.                         max {H(0, j)...S(i-1, j)} -q,
9.                         max {H(i, 0)...S(i, j-1)} -q,
10.                        0)
11.   best = max (H(i, j), best) }
```

Keterangan:

- $M, N$  = panjang masing-masing sekuens A dan B
- $H$  = array dua dimensi / matriks yang menyimpan nilai perhitungan
- $W(a_i, b_j)$  = nilai asosiasi antarsekuens
- $q$  = nilai gap

- best = variabel dengan nilai yang akan dimasukkan pada matriks H

### 2.6.3 Algoritma Smith-Waterman dan BLOSUM

Komputasi algoritma Smith-Waterman berdasarkan pada matriks subtitusi dan fungsi *gap penalty*. Keterkaitan algoritma Smith-Waterman dengan matriks subtitusi terletak pada asosiasi residu atau segmen sekuen pada elemen matriks subtitusi. Rumus Smith-Waterman yang berdasarkan pada fungsi *gap penalty* ditunjukkan pada persamaan 2.6 dan 2.7.

$$E_{ij} = \max \left[ \begin{array}{l} E_{i,j-1} - \text{gap}_{\text{ext}} \\ H_{i,j-1} - \text{gap}_{\text{open}} \end{array} \right] \quad (2.6)$$

$$F_{ij} = \max \left[ \begin{array}{l} F_{i-1,j} - \text{gap}_{\text{ext}} \\ H_{i-1,j} - \text{gap}_{\text{open}} \end{array} \right] \quad (2.7)$$

Persamaan 2.6 dan 2.7 menunjukkan perhitungan jika residu pada salah satu sekuens yang disejajarkan berakhir dengan *gap*. Rumus Smith-Waterman yang berdasarkan pada matriks subtitusi ditunjukkan pada rumus 2.8 sebagai berikut (Svetlin A. Manavski dan Giorgio Valle, 2008).

$$H_{ij} = \max \left[ \begin{array}{l} 0 \\ E_{ij} \\ F_{ij} \\ H_{i-1,j-1} + W(a_i, b_j) \end{array} \right] \quad (2.8)$$

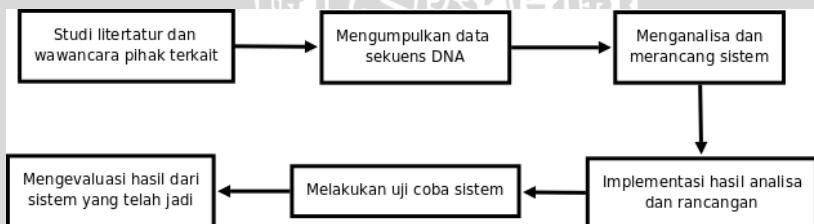
## BAB 3

### METODOLOGI DAN PERANCANGAN

Penelitian tentang pensejajaran sekuens dengan implementasi algoritma Smith-Waterman dalam menentukan tingkat kesamaan atau kecocokan dua buah sekuens ini harus mengikuti beberapa tahapan. Tahap-tahap untuk melakukan penelitian adalah sebagai berikut :

1. melakukan studi literatur dan wawancara dengan peneliti yang bergerak di bidang Bioinformatika,
2. mengumpulkan data sekuens DNA sebagai data uji melalui basis data yang telah disediakan oleh institusi terkait,
3. menganalisis dan melakukan perancangan sistem pensejajaran dengan menggunakan metode Smith-Waterman sesuai aturan yang telah ditentukan,
4. mengimplementasikan hasil analisa dan rancangan yang dilakukan pada tahap sebelumnya menjadi sistem pensejajaran sekuens,
5. melakukan uji coba sistem dengan menggunakan data yang telah dikumpulkan,
6. mengevaluasi hasil sistem dalam melakukan pensejajaran sekuens yang sesuai dengan aturan yang telah ditetapkan.

Gambar 3.1 menunjukkan bagan alur tahapan yang dilakukan pada penelitian ini.



Gambar 3.1. Bagan alur tahapan penelitian

#### 3.1 Sumber Data

Data sekuens DNA maupun asam amino telah disediakan melalui beberapa institusi yang terkait di bidang Bioinformatika. Data yang digunakan dalam skripsi ini akan diperoleh melalui situs <http://ncbi.nlm.nih.gov/GenBank>.

Format data sekuens berdasarkan format IUPAC (*International Union of Pure and Applied Chemistry*) (ditunjukkan pada Gambar 3.2). Setiap karakter/string data tersebut pada penelitian ini akan dikenal dengan istilah residu.

```
VITYENNNTWNQTYVNISNTNFAAGQSVSVKLAGNSSLCPV  
SGWAIYSKDNDNSVRIGSKGDVFVIREPFISCSPLECRTFFLTQ  
GALLNDKHSNGTIKDRSPTDTIKSWRNNILRTQESECACVNG  
SREPFIASCSPLECRTFFLTQGALLNDKHSNGTIKDRSWS
```

Gambar 3.2. Format IUPAC

### 3.2 Struktur Data

Data uji coba yang akan sering ditemukan berupa *string* sehingga dalam pengujian akan digunakan struktur data array. Struktur data array memiliki metode pemberian indeks sehingga data dapat diakses sesuai dengan indeksnya.

Pada penelitian ini, akan digunakan tiga macam struktur data yaitu array berdimensi satu dengan tipe data *string* untuk inisialisasi awal, array berdimensi dua/matriks untuk proses pensejajaran, dan hash table untuk asosiasi residu.

#### 3.2.1 Inisialisasi awal

Inisialisasi awal untuk metode Smith-Waterman menggunakan array berdimensi satu dengan tipe data *string*.

#### 3.2.2 Proses pensejajaran

Proses pensejajaran lebih banyak menggunakan struktur data array berdimensi dua atau yang sering disebut dengan matriks. Matriks ini digunakan untuk skor perhitungan dan pointer.

#### 3.2.3 Asosiasi residu

Asosiasi residu merupakan nilai yang diambil dari sebuah matriks substitusi. Setiap asosiasi memiliki kunci dan nilai spesifik sehingga tabel *hash* lebih tepat digunakan untuk menyimpan data tersebut. Tabel hash yang digunakan adalah

tabel *hash* yang dinamis sehingga penggunaan ruang memori lebih efisien.

### 3.3 Deskripsi Sistem

Sistem yang dirancang merupakan sistem pensejajaran sekuens dengan mengimplementasikan metode Smith-Waterman dalam menentukan tingkat kesamaan atau kecocokan antara dua buah sekuens yang ditinjau dari tingkat kesamaan kodon pembentuk asam amino. Sistem pensejajaran memerlukan data masukan (*input*) berupa sekuens DNA ataupun asam amino.

Data masukan menggunakan standar IUPAC karena format tersebut hanya meliputi sekuens DNA atau asam amino tanpa mengikutkan informasi lain dari sebuah individu atau spesies. Sistem akan melakukan verifikasi sebelum melakukan pensejajaran terhadap data masukan. Jika data masukan berupa sekuens DNA maka akan dilakukan proses penerjemahan menjadi sekuens asam amino. Jika data masukan berupa sekuens asam amino, maka sistem akan melakukan pensejajaran dengan menggunakan metode Smith-Waterman.

Data keluaran (*output*) yang dihasilkan oleh sistem berupa hasil pensejajaran sekuens dengan menampilkan tingkat kesamaan sekuens tersebut sesuai dengan tingkat substitusi mutasi asam amino.

### 3.4 Perancangan Sistem

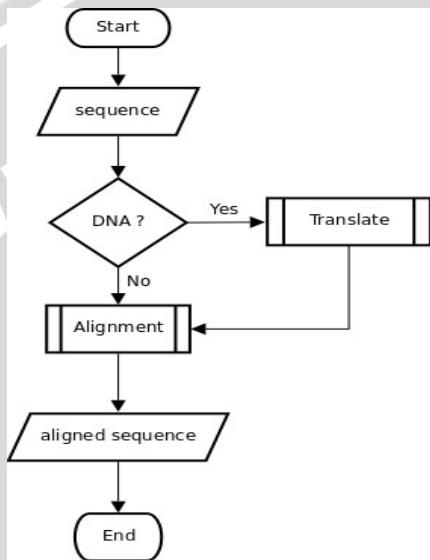
Perancangan sistem pensejajaran sekuens meliputi beberapa tahapan rancangan sebagai berikut.

#### 3.4.1 Rancangan proses

Sistem pensejajaran sekuens dengan mengimplementasikan metode Smith-Waterman memiliki beberapa rangkaian proses yang harus dikerjakan. Proses-proses yang harus dilakukan oleh sistem tersebut adalah sebagai berikut.

- 1) Pengguna harus memasukkan data sekuens yang berupa string sesuai dengan format yang telah ditentukan.
- 2) Sistem melakukan verifikasi data masukan, apakah masukan berupa sekuens DNA atau asam amino.

- 3) Jika data masukan berupa sekuen DNA, maka sekuen akan melakukan penerjemahan DNA menjadi asam amino.
- 4) Sistem melakukan proses pensejajaran terhadap data masukan.
- 5) Data keluaran berupa data sekuen yang telah disejajarkan.



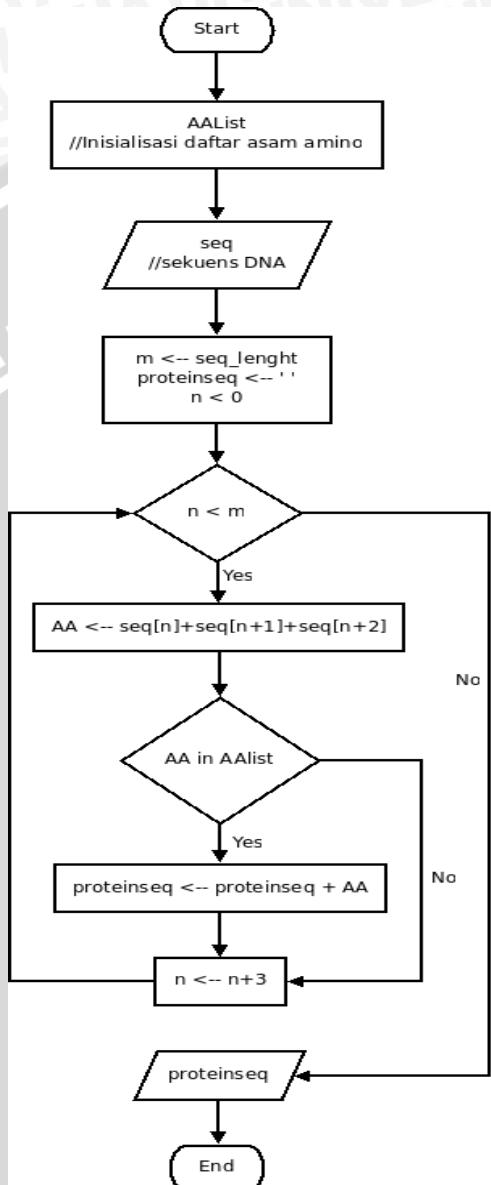
Gambar 3.3. Diagram alir proses sistem pensejajaran sekuen

#### 3.4.1.1 Proses penerjemahan (*translate*)

Proses penerjemahan yang dilakukan hanya menggunakan *single frame* (rangkaian tunggal). Penerjemahan dilakukan dengan menjalankan proses sebagai berikut.

1. Membuat list yang berisikan aturan (*rule*) penentuan asam amino berdasarkan struktur kodonnya.
2. Mencari kecocokan kodon secara *looping* pada data masukan dengan kodon yang terdapat pada list yang telah dibuat.
3. Keluaran berupa data sekuen asam amino.

Penjelasan di atas akan secara jelas dideskripsikan pada gambar 3.4.



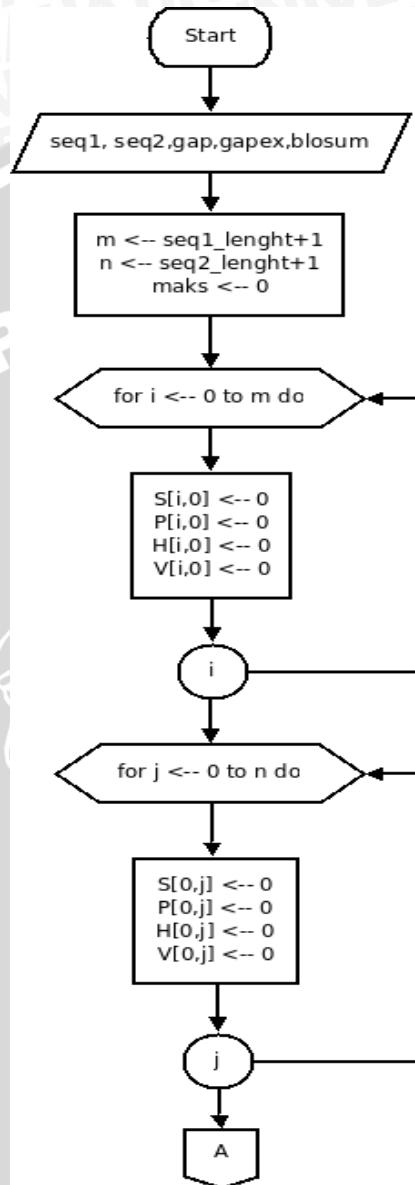
Gambar 3.4. Diagram alir proses penerjemahan

### 3.4.1.2 Proses Pensejajaran (*alignment*)

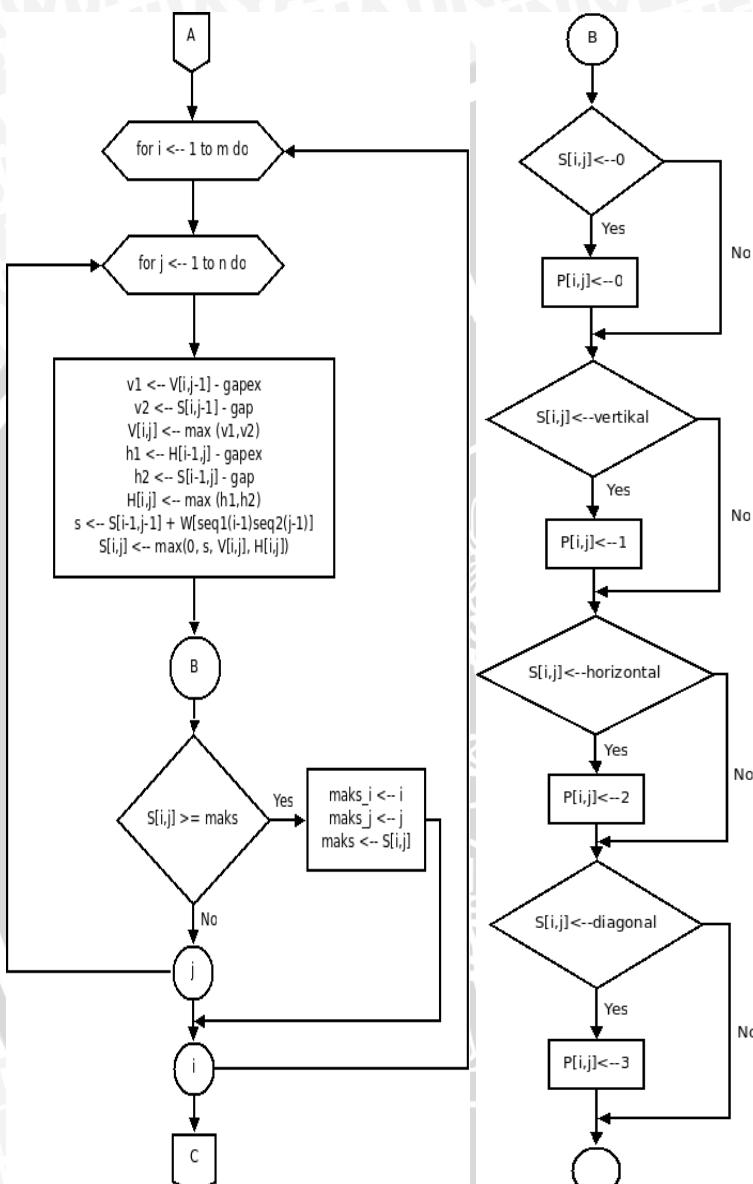
Proses pensejajaran yang dilakukan oleh sistem akan dibagi menjadi beberapa bagian. Bagian-bagian tersebut adalah sebagai berikut.

1. Inisialisasi awal (ditunjukkan pada gambar 3.5)
  - a. Menentukan panjang sekuens pertama dan kedua. Panjang tersebut akan menjadi indeks perulangan pada metode Smith-Waterman.
  - b. Mendeklarasikan matriks S untuk perhitungan, matriks V dan H untuk *gap* dan matriks P untuk pointer *traceback*. Matriks tersebut memiliki ordo sesuai dengan panjang masing-masing sekuens.
  - c. Menginisialisasi baris dan kolom dengan nilai 0
2. Menjalankan metode Smith-Waterman (ditunjukkan pada gambar 3.6(a))
  - a. Mencari nilai variabel tetangga elemen matriks S pada posisi diagonal, vertikal, dan horizontal.
  - b. Menentukan nilai maksimum dari nilai tetangga elemen matriks H.
3. Menentukan pointer (ditunjukkan pada gambar 3.6(b))
4. Melakukan penelusuran balik (ditunjukkan pada gambar 3.7)
  - a. Penelusuran dilakukan untuk mencari nilai terbesar dari tetangga pasangan residu pada posisi diagonal, vertikal, dan horizontal.
  - b. Pergerakan pointer dimulai dari posisi indeks terbesar.
  - c. Proses penelusuran akan berhenti ketika pointer menunjukkan nilai nol.
5. Menjalankan proses *result*

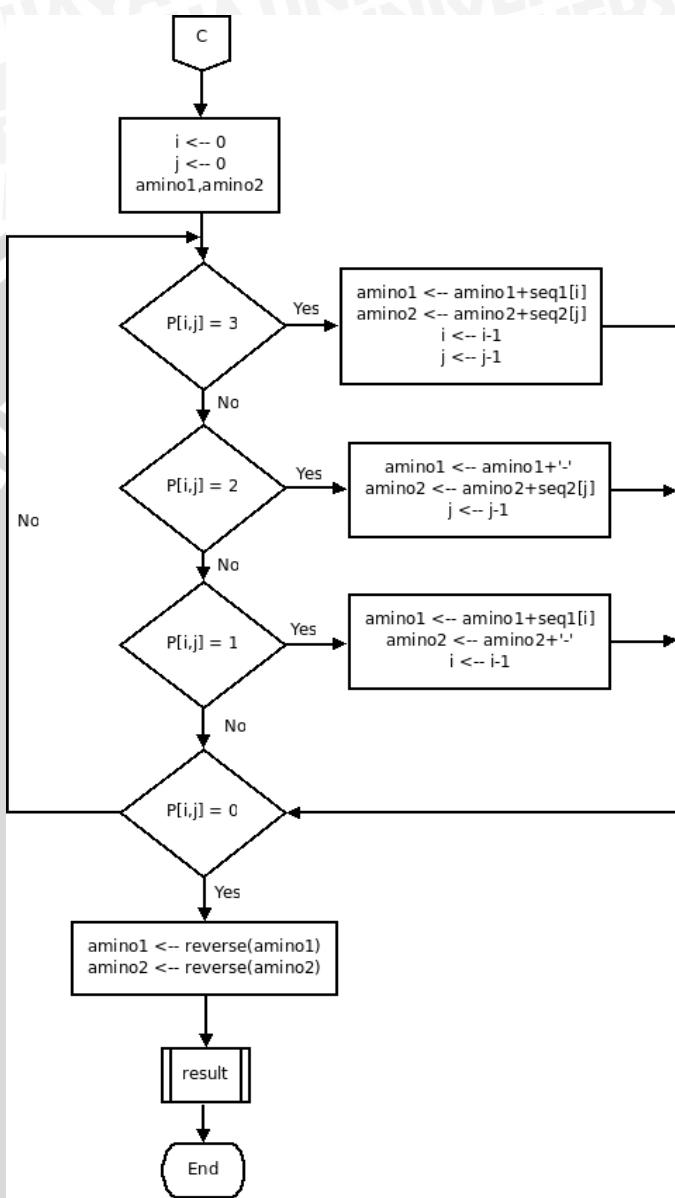
Proses result akan dijelaskan pada subbab 3.4.1.3.



Gambar 3.5. Diagram alir inisialisasi awal proses pencejajaran



Gambar 3.6. Diagram alir proses



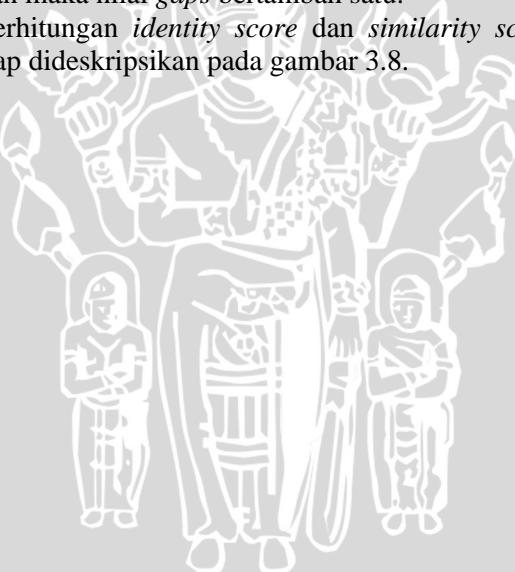
Gambar 3.7. Diagram alir penelusuran balik (*traceback*)

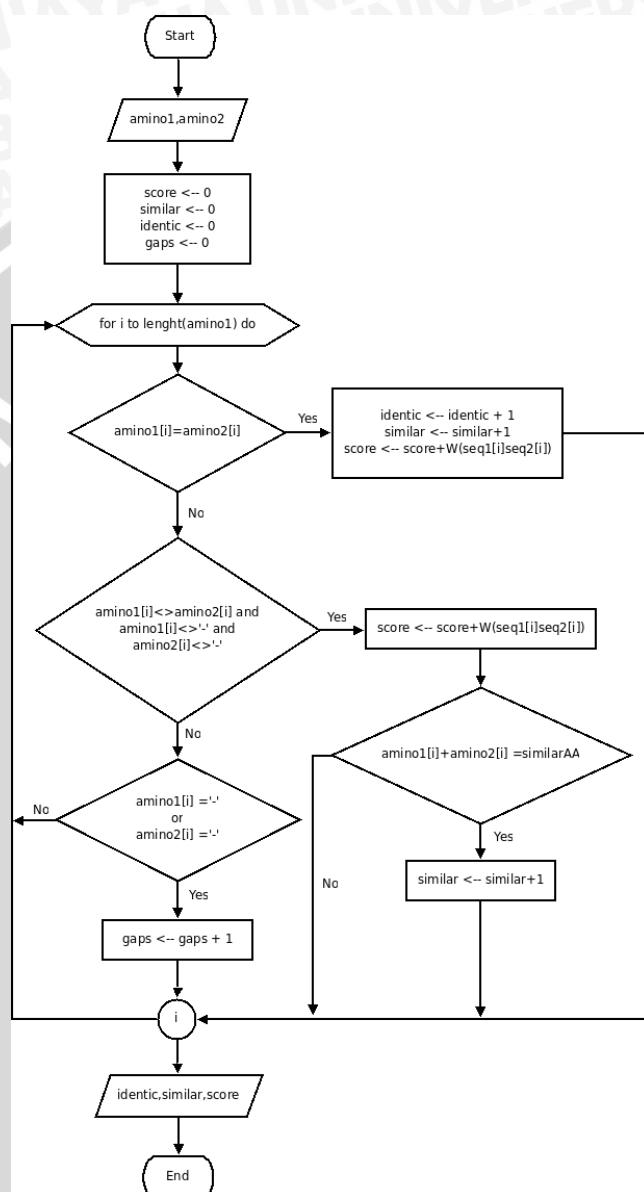
### 3.4.1.3 Proses perhitungan *identity score* dan *similarity score* (*result*)

*Identity score* dan *similarity score* residi dihitung sesuai dengan nilai pada matriks substitusi BLOSUM. Adapun proses perhitungan dijelaskan sebagai berikut.

1. Jika pasangan residu sama maka *identity score* dan *similarity score* masing-masing bertambah satu dan skor bertambah sesuai dengan nilai pasangan residu pada matriks substitusi.
2. Jika pasangan residu tidak sama tetapi memiliki kemiripan maka *similarity score* bertambah satu dan skor bertambah sesuai dengan nilai pasangan residu pada matriks substitusi.
3. Jika pasangan residu tidak sama dan tidak memiliki kemiripan maka nilai *gaps* bertambah satu.

Proses perhitungan *identity score* dan *similarity score* secara lengkap dideskripsikan pada gambar 3.8.



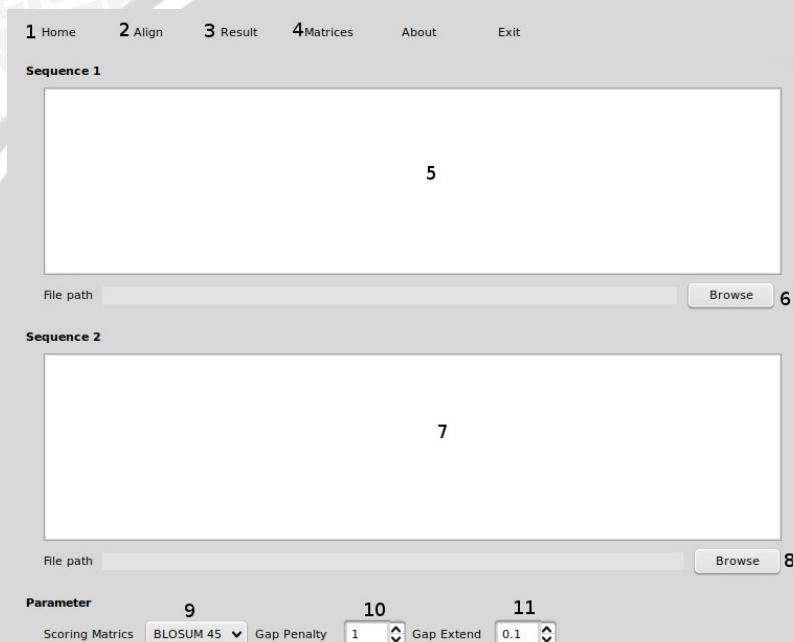


Gambar 3.8. Diagram alir proses perhitungan *identity score* dan *similarity score*

### 3.4.2 Rancangan Antarmuka Pengguna

#### 3.4.2.1 Rancangan antarmuka masukan

Antarmuka masukan dimaksudkan untuk tempat memasukkan data uji. Data masukan bisa berupa berkas teks atau data masukan manual (ditunjukkan pada gambar 3.9).



Gambar 3.9. Rancangan antarmuka masukan

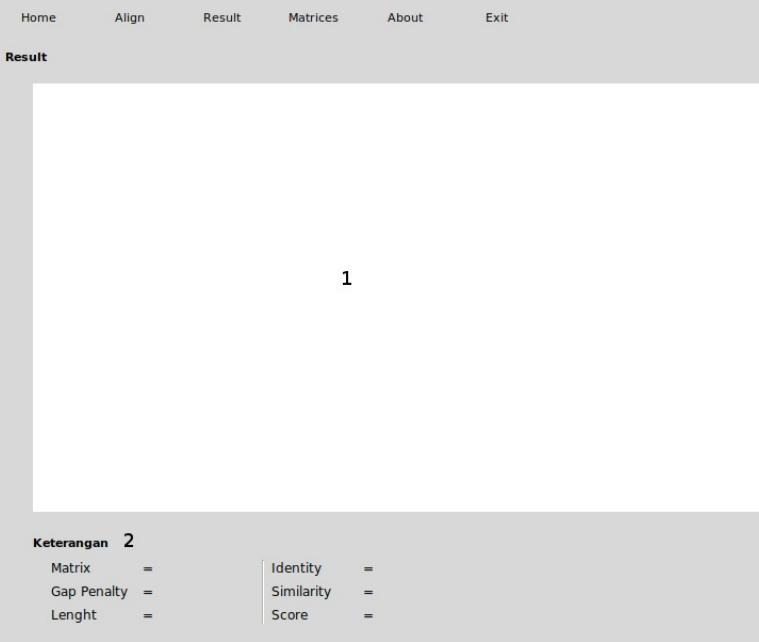
Keterangan gambar 3.9 :

1. Tombol 'HOME' untuk menampilkan halaman masukan
2. Tombol 'ALIGN' untuk menjalankan proses pensejajaran
3. Tombol 'Result' untuk menampilkan halaman hasil
4. Tombol 'Matrices' untuk membuka matriks dan menampilkan halaman matriks.
5. PlaintextBox untuk masukan sekuens pertama.
6. Tombol 'BROWSE' untuk membuka file.
7. PlaintextBox untuk masukan sekuens kedua.
8. Tombol 'BROWSE' untuk membuka file.

9. ComboBox untuk menentukan matriks substitusi
10. SpinBox untuk menentukan nilai gap open
11. SpinBox untuk menentukan nilai gap extend

### 3.4.2.2 Rancangan antarmuka hasil

Antarmuka hasil bertujuan untuk menampilkan hasil pensejajaran data masukan. Hasil pensejajaran akan dilengkapi dengan beberapa keterangan penting (ditunjukkan pada gambar 3.10).



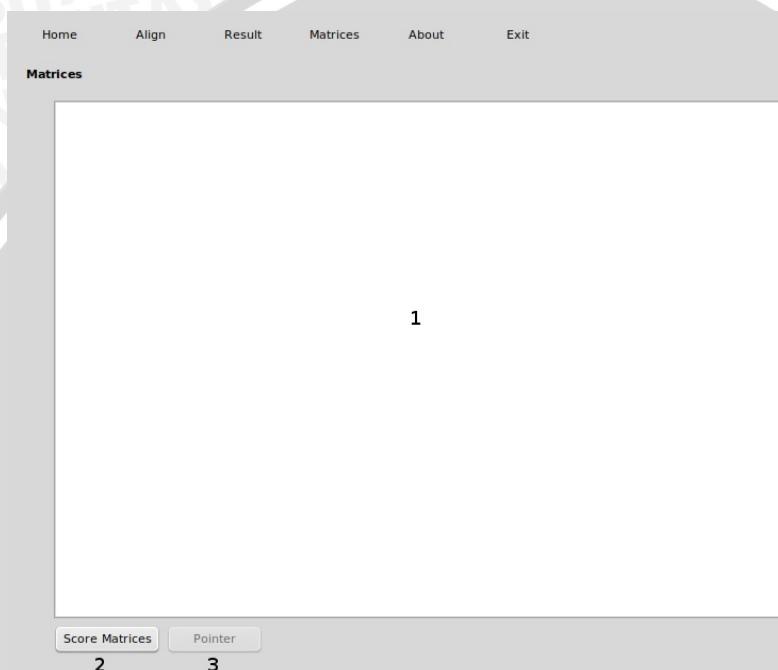
Gambar 3.10. Rancangan antarmuka hasil

Keterangan gambar 3.10 :

1. PlaintextBox untuk menampilkan hasil pensejajaran
2. Keterangan untuk menampilkan hasil perhitungan

### 3.4.2.3 Rancangan antarmuka data matriks dan pointer

Antarmuka ini bertujuan untuk menampilkan matriks skor dan pointer sebagai rujukan atau gambaran proses yang dilakukan oleh sistem (ditunjukkan pada gambar 3.11).



Gambar 3.11. Rancangan antarmuka matriks skor dan pointer

Keterangan gambar 3.11 :

1. TableWidget untuk menampilkan matriks skor dan pointer
2. Tombol 'Score Matrics' untuk menampilkan halaman matriks skor
3. Tombol 'Pointer' untuk menampilkan halaman pointer

## 3.5 Rancangan Uji Coba

Uji coba sistem akan dilakukan untuk mengetahui kemampuan sistem dalam melakukan proses yang telah

dirancang sebelumnya. Uji coba yang dilakukan berdasarkan parameter-parameter sebagai berikut.

a. *Gap penalty*

*Gap penalty* yang digunakan adalah *gap open penalty* dan *gap extend penalty*. Nilai *gap extend* selalu lebih kecil dari *gap open*.

b. *Substitution matrices*

*Substitution matrices* merupakan matriks subtitusi dengan berpatokan pada matriks subtitusi BLOSUM. Pada penelitian ini akan digunakan 3 jenis matriks subtitusi sebagai perbandingan yaitu :

- BLOSUM45
- BLOSUM62
- BLOSUM80

Parameter yang disebutkan di atas akan digunakan untuk mengetahui hasil perhitungan variabel berikut ini.

a. *Identity score* (Skor identik)

*Identity score* adalah skor untuk tingkat kesamaan yang identik. Kesamaan yang identik seperti Alanine (A) sama dengan Alanine (A). Nilai skor identik didapatkan dengan persamaan 3.1.

$$\text{Identity} = \left( \sum_{\text{identikAA}} / \text{Aligned}_{\text{length}} \right) \times 100 \% \quad (3.1)$$

Nilai identity score adalah jumlah kesamaan residu yang terdapat pada hasil

b. *Similarity / Positive score* (Skor kemiripan)

*Similarity / positive score* merupakan skor untuk tingkat kesamaan yang berdasarkan pada kemiripan dua residu. Contoh kemiripan dua residu seperti Isoleucine (I) mirip dengan Methionine (M). Nilai skor kemiripan didapatkan dengan persamaan 3.2.

$$\text{Similarity} = \left( \sum_{(\text{similarAA} + \text{identikAA})} / \text{Aligned}_{\text{length}} \right) \times 100 \% \quad (3.2)$$

c. *Total Score* (Skor keseluruhan)

*Total score* merupakan nilai terbesar dalam matriks

perhitungan. *Total score* menentukan awal dari pergerakan pointer *traceback*.

### 3.5.1 Uji coba perhitungan manual

Untuk mengetahui bagaimana sistem melakukan kerjanya, maka diperlukan uji coba perhitungan manual. Sebagai contoh, diambil dua data uji coba berupa sekuens asam amino dari buah Kiwi dan buah Pepaya.

Data masukan :

a = Sekuens1 (Kiwi) : VDHAIIVIV  
b = Sekuens2 (Pepaya) : VDHAVAAG

Inisialisasi awal :

- gap open (g)= 1
- gap extend = 0.1
- match score ( $w(a_i, b_j)$ ) = nilai pasangan residu pada matriks substitusi
- matriks substitusi=BLOSUM62
- matriks  $S[i,j]$ ,  $V[i,j]$ ,  $H[i,j]$
- pointer  $P[i,j]$
- $S[0,j] \& S[i,0] = 0$
- $V[0,j] \& V[i,0] = 0$
- $H[0,j] \& H[i,0] = 0$
- $P[0,j] \& P[i,0] = 0$

Rumus perhitungan skor sekuens ditunjukkan pada persamaan 3.3.

$$S_{ij} = \max \begin{cases} 0 \\ \max(S_{i-1, j-1} - \text{gap}, V_{i-1, j-1} - \text{gapex}) \\ \max(S_{i-1, j} - \text{gap}, H_{i-1, j} - \text{gapex}) \\ S_{i-1, j-1} - W(a_i, b_j) \end{cases} \quad (3.3)$$

Hasil perhitungan manual dapat dilihat pada Lampiran 1. Berdasarkan perhitungan manual yang dilakukan maka

diperoleh hasil seperti yang ditunjukkan pada tabel 3.1.

Tabel. 3.1. Matriks hasil perhitungan

	0	1 (V)	2 (D)	3 (H)	4 (A)	5 (V)	6 (A)	7 (A)	8 (V)
0	0	0	0	0	0	0	0	0	0
1 (V)	0	4	3	2,9	2,8	4	3	2,9	4
2 (D)	0	3	10	9	8,9	8,8	8,7	8,6	8,5
3 (H)	0	2,9	9	18	17	16,9	16,8	16,7	16,6
4 (A)	0	2,8	8,9	17	22	21	20,9	20,8	20,7
5 (I)	0	3	8,8	16,9	21	25	24	23,9	23,8
6 (V)	0	4	8,7	16,8	20,9	25	25	24	27,9
7 (I)	0	4	8,6	16,7	20,8	24	24	24	27
8 (V)	0	4	8,5	16,6	20,7	24,8	24	24	28

Berdasarkan matriks hasil perhitungan di atas (Tabel 3.1) maka hasil pensejajaran *a* dengan *b* ditunjukkan pada gambar 3.12.

VDHAIIVIV  
||||:...|  
VDHAVA AV

Gambar 3.12. Hasil uji coba perhitungan manual

Hasil perhitungan manual yang dilakukan di atas menunjukkan gambaran data sebagai berikut.

$$\begin{aligned} \text{Identity score} &= 5/8 \\ \text{Similarity score} &= 6/8 \\ \text{Gaps} &= 0/8 \\ \text{Total score} &= 4+6+8+4+3+0-1+4 \\ &= 28 \end{aligned}$$

Nilai *similarity score* lebih besar dibandingkan *identity score* karena pada dua sekuen tersebut terdapat asam amino/residu yang dapat dikatakan mirip (asam amino I dengan V) sehingga dapat dikatakan Isoleucine (I) dapat digantikan oleh Valine (V) pada saat terjadinya mutasi.

### 3.5.2 Rancangan analisis

Analisis pencejajaran dilakukan sesuai dengan parameter rancangan uji coba yang telah disebutkan pada subbab 3.5 sebelumnya. Analisis sistem pencejajaran dilakukan dengan tiga tahap pengujian sebagai berikut.

#### 3.5.2.1 Analisis hubungan antara gap penalty dengan *identity score* dan *similarity score*

Pengujian dimulai dari menggunakan *gap open* dan *gap extend* minimum sampai pada nilai maksimum. Pengujian ini dimaksudkan untuk mengetahui seberapa besar perubahan skor kesamaan dan skor kemiripan terhadap perubahan *gap open* dan *gap extend*. Pengujian dilakukan pada data uji yang sama. Tabel 3.2 dan 3.3 menunjukkan rancangan tabel hubungan antara *gap open* dengan *identity score* dan *similarity score*.

Tabel 3.2. Tabel hubungan *gap open* dengan *identity score*

Gap Open	<i>identity score (%)</i>		
	BLOSUM45	BLOSUM62	BLOSUM80
1			
2			
...			
n			

Tabel 3.3. Tabel hubungan *gap open* dengan *similarity score*

Gap Open	<i>similarity score (%)</i>		
	BLOSUM45	BLOSUM62	BLOSUM80
1			
2			
...			
n			

Tabel 3.4 dan 3.5 masing-masing menunjukkan tabel

hubungan antara gap extend dengan *identity score* dan *similarity score*

Tabel 3.4. Tabel hubungan *gap extend* dengan *identity score*

Gap Extend	<i>identity score (%)</i>		
	BLOSUM45	BLOSUM62	BLOSUM80
0.1			
0.2			
...			
n			

Tabel 3.5. Tabel hubungan *gap extend* dengan *similarity score*

Gap Extend	<i>similarity score (%)</i>		
	BLOSUM45	BLOSUM62	BLOSUM80
0.1			
0.2			
...			
n			

### 3.5.2.2 Analisis hubungan antara *gap penalty* dengan kemunculan *gap* pada hasil pensejajaran

Pengujian tahap kedua dilakukan dengan menggunakan data yang sama pada pengujian tahap pertama. Tahap pengujian ini akan memperhatikan hubungan antara *gap penalty* dengan kemunculan *gap* pada hasil pensejajaran. Tabel 3.6 dan 3.7 menunjukkan rancangan tabel hubungan antara *gap open* dengan *identity score* dan *similarity score*.

Tabel 3.6. Tabel hubungan *gap open* dengan kemunculan *gap* pada hasil pensejajaran

Gap Open	Kemunculan gap (%)		
	BLOSUM45	BLOSUM62	BLOSUM80
1			
2			
...			
n			

Tabel 3.7. Tabel hubungan *gap open* dengan kemunculan *gap* pada hasil pensejajaran

Gap Extend	Kemunculan gap (%)		
	BLOSUM45	BLOSUM62	BLOSUM80
0.1			
0.2			
...			
n			

### 3.5.2.3 Analisis hubungan matriks substitusi dengan *identity score* dan *similarity score*

Pengujian ketiga dilakukan dengan menggunakan sejumlah  $N$  data acak. Pensejajaran dilakukan dengan menggunakan nilai *gap open* dan *gap extend* yang sama untuk setiap data. Penelitian ini bertujuan untuk melihat letak hubungan matriks substitusi terhadap *identity score* dan *similarity score*. Tabel 3.8 dan 3.9 menunjukkan rancangan tabel hubungan matriks substitusi dengan *identity score* dan *similarity score*.

Tabel 3.8. Tabel hubungan matriks substitusi dengan *identity score*

Data Uji Coba	<i>identity score (%)</i>		
	BLOSUM45	BLOSUM62	BLOSUM80
1			
2			
...			
N			

Tabel 3.9. Tabel hubungan matriks substitusi dengan *similarity score*

Data Uji Coba	<i>similarity score (%)</i>		
	BLOSUM45	BLOSUM62	BLOSUM80
1			
2			
...			
N			

### 3.5.3 Rancangan validasi hasil

Validasi hasil dimaksudkan untuk mengetahui keakuriasan hasil dari sistem pensejajaran. Validasi akan dilakukan dengan cara membandingkan hasil pensejajaran oleh sistem dibuat dengan hasil pensejajaran oleh sistem yang telah ada. Dalam penelitian ini, perbandingan akan dilihat pada *identity score*, *similarity score*, dan jumlah kemunculan gap pada hasil pensejajaran.

Data untuk pengukuran validasi hasil pensejajaran diambil dari hasil pensejajaran oleh sistem EMBOSS. Validasi yang dilakukan terhadap data mengikuti parameter default dari sistem EMBOSS. Sistem EMBOSS merupakan sistem pensejajaran yang berada di situs <http://www.ebi.ac.uk/Tools/emboss/align/index.html>.

UNIVERSITAS BRAWIJAYA



## BAB 4

# IMPLEMENTASI DAN PEMBAHASAN

### 4.1 Lingkungan Implementasi

Lingkungan implementasi sistem pensejajaran sekuens asam amino secara garis besar menggunakan perangkat keras dengan spesifikasi sebagai berikut.

1. Prosesor Intel Celeron M
2. Memori DDR2 1 GB
3. VGA Intel GMA 950
4. Harddisk 80 GB

Sistem ini juga menggunakan perangkat lunak sebagai berikut.

1. Sistem operasi Ubuntu Linux 9.04
2. Bahasa pemrograman Python 2.6
3. Eclipse Software Development Kit 3.5.0
4. Qt4 Graphical User Interface Designer

### 4.2 Implementasi Sistem

Pada subbab 3.4 telah dijelaskan bagaimana rancangan sistem yang akan dibangun. Penjelasan berikut ini adalah pemaparan secara menyeluruh tentang implementasi sistem yang telah dirancang.

#### 4.2.1 Implementasi proses penerjemahan

Proses penerjemahan dilakukan ketika data masukan berupa sekuens DNA. Data masukan akan diverifikasi oleh sebuah prosedur yang bertipe *boolean*. Jika nilai kembalian dari prosedur adalah TRUE, maka data masukan berupa DNA dan jika FALSE maka data masukan berupa asam amino.

Prosedur untuk verifikasi data masukan ditunjukkan pada gambar 4.1.

```

def isDna (seq):
    dna = seq[0:10]
    m = len(dna)
    yes,no = 0,0
    for i in range (m):
        if dna[i] in 'ATGC':
            yes = yes+1
        else:
            no = no+1
    if yes > no :
        return True
    else:
        return False
    return True

```

Gambar 4.1. Prosedur verifikasi data masukan

Prosedur penerjemahan data masukan ditunjukkan pada gambar 4.2.

```

def translate_dna(sequence):
    gencode = {
        'ATA': 'I', 'ATC': 'I', 'ATT': 'I', 'ATG': 'M', 'ACA': 'T',
        'ACC': 'T', 'ACG': 'T', 'ACT': 'T', 'AAC': 'N', 'AAT': 'N',
        'AAA': 'K', 'AAG': 'K', 'AGC': 'S', 'AGT': 'S', 'AGA': 'R',
        'AGG': 'R', 'CTA': 'L', 'CTC': 'L', 'CTG': 'L', 'CTT': 'L',
        'CCA': 'P', 'CCC': 'P', 'CCG': 'P', 'CCT': 'P', 'CAC': 'H',
        'CAT': 'H', 'CAA': 'Q', 'CAG': 'Q', 'CGA': 'R', 'CGC': 'R',
        'CGG': 'R', 'CGT': 'R', 'GTA': 'V', 'GTC': 'V', 'GTG': 'V',
        'GTT': 'V', 'GCA': 'A', 'GCC': 'A', 'GCG': 'A', 'GCT': 'A',
        'GAC': 'D', 'GAT': 'D', 'GAA': 'E', 'GAG': 'E', 'GGA': 'G',
        'GGC': 'G', 'GGG': 'G', 'GGT': 'G', 'TCA': 'S', 'TCC': 'S',
        'TCG': 'S', 'TCT': 'S', 'TTC': 'F', 'TTT': 'F', 'TTA': 'L',
        'TTG': 'L', 'TAC': 'Y', 'TAT': 'Y', 'TAA': '*', 'TAG': '*',
        'TGC': 'C', 'TGT': 'C', 'TGA': '*', 'TGG': 'W',
    }
    proteinseq = ''
    for n in range(0,len(sequence),3):
        if gencode.has_key(sequence[n:n+3]) == True:
            proteinseq += gencode[sequence[n:n+3]]
    return proteinseq

```

Gambar 4.2. Prosedur penerjemahan

Prosedur penerjemahan menggunakan *hash table* sebagai tabel yang menyimpan informasi kodon atau pasangan basa dengan asam amino yang dihasilkannya. Kunci pada tabel hash adalah kodon dan nilainya adalah asam amino bentukan kodon.

## 4.2.2 Implementasi proses pensejajaran

Proses pensejajaran menggunakan beberapa parameter yang telah diatur pada saat melakukan inisiasi data masukan. Parameter yang digunakan seperti matriks substitusi, *gap open*, dan *extend gap*.

Pensejajaran sekuens melewati beberapa prosedur sebelum mendapatkan hasil pensejajaran. Adapun prosedur yang dimaksud sebagai berikut.

### 4.2.2.1 Inisialisasi

Proses inisialisasi dimaksudkan untuk menentukan atau mendeklarasikan beberapa variabel yang merepresentasikan panjang masing-masing data masukan, menentukan matriks substitusi yang digunakan, nilai *gap penalty*, dan membuat matriks dengan ordo  $m \times n$ . Proses inisialisasi ditunjukkan pada gambar 4.3.

```
self.gap = self.ui.sbGap.value()
self.gapex = self.ui.sbGapex.value()
self.getBlosom(self.ui.cbBlosum.currentIndex())
m = len(self.seq1)+1
n = len(self.seq2)+1
maks = 0

v1,v2,h1,h2,s = 0,0,0,0,0

S = zeros((m,n))
P = zeros((m,n))
H = zeros((m,n))
V = zeros((m,n))
```

Gambar 4.3. Inisialisasi

Matriks S, P, V, dan H pada inisialisasi masing-masing merepresentasikan matriks skor, matriks pointer, matriks untuk pointer vertikal, dan matriks untuk pointer horizontal.

### 4.2.2.2 Implementasi metode Smith-Waterman

Proses pensejajaran dengan metode Smith-Waterman, menggunakan tiga matriks yang telah dideklarasikan pada saat inisialisasi. Matriks S menyimpan nilai skor perhitungan, matriks V dan H menyimpan skor berdasarkan kemunculan gap pada sekuens. Metode tersebut ditunjukkan pada gambar 4.4.

```

for i in range(m):
    S[i,0] = 0
    H[i,0] = 0
    V[i,0] = 0
for j in range(n):
    S[0,j] = 0
    H[0,j] = 0
    V[0,j] = 0
for i in range(1,m):
    for j in range(1,n):
        v1 = V[i-1,j] - self.gapex
        v2 = S[i-1,j] - self.gap
        V[i,j] = max(v1,v2)

        h1 = H[i,j-1] - self.gapex
        h2 = S[i,j-1] - self.gap
        H[i,j] = max(h1,h2)
        pair = self.seq1[i-1]+self.seq2[j-1]
        if self.blos.has_key(pair):
            s = S[i-1,j-1] + self.blos[pair]
            S[i,j] = max(0,s,H[i,j],V[i,j])

```

Gambar 4.4. Metode Smith-Waterman

#### 4.2.2.3 Penentuan pointer

Pointer dimaksudkan untuk mengetahui arah penelusuran hasil pensejajaran. Nilai pointer disimpan dalam bentuk matriks. Penentuan pointer ditunjukkan pada gambar 4.5.

```

if S[i,j]==0:
    P[i,j]=0 #pointer untuk berhenti
if S[i,j]==V[i,j]:
    P[i,j]=1 #pointer untuk ke atas
if S[i,j]==H[i,j]:
    P[i,j]=2 #pointer untuk ke kiri
if S[i,j]==S[i,j]:
    P[i,j]=3 #pointer untuk ke diagonal

if S[i,j] > maks:
    maks_i = i
    maks_j = j
    maks = S[i,j]

```

Gambar 4.5. Penentuan pointer

Gambar 4.5 menunjukkan empat macam posisi pointer yang nantinya akan mengatur pergerakan pointer tersebut. Nilai nol berarti pointer berhenti. Nilai 1 (satu) menandakan bahwa pointer nantinya akan bergerak ke atas. Nilai 2 (dua) menandakan pointer bergerak ke kiri dan nilai 3 (tiga) menandakan pointer bergerak ke arah diagonal.

#### 4.2.2.4 Implementasi proses penelusuran (*traceback*)

Penelusuran (traceback) dilakukan setelah matriks pointer telah ditentukan. Pergerakan pointer dimulai dari indeks pada matriks skor yang memiliki nilai terbesar. Misalnya nilai terbesar dari matriks skor terletak di indeks ( $i=8, j=7$ ), maka penelusuran dilakukan mulai dari indeks tersebut sampai pada nilai pointer menunjukkan angka nol. Pointer bergerak dari kanan ke kiri sesuai dengan nilai pada pointer yang ditentukan sebelumnya. Implementasi proses penelusuran ditunjukkan pada gambar 4.6.

```
i,j = maks_i,maks_j

while P[i,j]!=0:
    if P[i,j]==3:
        self.align1 += self.seq1[i-1]
        self.align2 += self.seq2[j-1]
        i -= 1
        j -= 1
    elif P[i,j]==2:
        self.align1 += '_'
        self.align2 += self.seq2[j-1]
        j -= 1
    else:
        self.align1 += self.seq1[i-1]
        self.align2 += '_'
        i -= 1

self.align1 = self.align1[::-1]
self.align2 = self.align2[::-1]
```

Gambar 4.6. Implementasi proses penelusuran

Penelusuran yang ditunjukkan pada gambar 4.6 di atas memberikan hasil pensejajaran (*self.align1* dan *self.align2*) berupa sebuah *string* yang susunannya terbalik. Susunan *string* tersebut diputar dengan cara mengubah indeks awal pada string menjadi indeks terakhir. Cara tersebut ditunjukkan pada baris kode *self.align1 = self.align1[::-1]* dan *self.align2 = self.align2[::-1]*.

#### 4.2.2.5 Implementasi proses perhitungan *identity score* dan *similarity score (result)*

Rumus perhitungan *identity score* dan *similarity score* telah ditunjukkan pada persamaan 3.1 dan 3.2. *Identity score* didapatkan dari jumlah nilai matriks substitusi terhadap pasangan residu yang sama secara identik. *Similarity score* didapatkan dari *identity score* ditambah dengan jumlah nilai matriks substitusi terhadap pasangan residu yang memiliki kemiripan. Seperti yang telah dibahas sebelumnya, kemiripan sebuah pasangan residu dilihat dari nilai bulat positif pada matriks substitusi. Oleh karena itu, sebuah prosedur dibuat untuk menyimpan daftar semua pasangan residu dengan nilai bulat positif yang ditunjukkan pada gambar 4.7.

```
def similarAA(self):
    similar = []
    AA = 'ARNDQCEGHILKMFPTWYV'

    for i in range(1,len(AA)+1):
        for j in range(1,len(AA)+1):
            pair = AA[i-1]+AA[j-1]
            if self.blos[pair] > 0 and AA[i-1]!=AA[j-1]:
                similar.append(pair)
    return similar
```

Gambar 4.7. Prosedur membuat daftar asam amino yang mirip

Perhitungan *identity score* dan *similarity score* ditunjukkan pada gambar 4.8.

```
for i in range(n):
    pair = self.align1[i]+self.align2[i]
    if self.align1[i]==self.align2[i]:
        simbol += '|'
        iden += 1
        simi += 1
    elif self.align1[i]!=self.align2[i] and self.align1[i]!='-' and self.align2[i]!='-':
        ln = len(sim)
        for j in range(ln):
            found = 0
            if pair == sim[j]:
                found = 1
                break
        if found==1:
            simbol += ':'
            simi += 1
        if found==0:
            simbol += '.'
    elif self.align1[i]=='-' or self.align2[i]=='-':
        simbol += '_'
        gaps += 1
```

Gambar 4.8 Perhitungan *identity score* dan *similarity score*

### 4.3 Implementasi Antarmuka Pengguna

Antarmuka pengguna dirancang sesuai dengan fungsi untuk menampilkan hasil pensejajaran sekuens. Secara garis besar, antarmuka pengguna yang dirancang meliputi beberapa bagian halaman dan komponen sebagai berikut.

#### 4.3.1 Halaman *Home*

Halaman *Home* berisikan komponen untuk data masukan dan beberapa parameter. Parameter yang dimaksudkan adalah *checkbox* untuk pilihan matriks substitusi, *spinbox* untuk *gap open* dan *gap extend*. Halaman *Home* ditunjukkan pada gambar 4.9.

Home      History      Align      Result      Matrices      About      Exit

### Sequence 1

```
VITYENNTWVNQTYVNIISNTNFAAGQSVVSVKLGNSSLCPVSGWAIYSKDNVRIGSKGDVFVIREPFISCSPLECRTFFLTO
GALLNDKHSNGTIKDRSPTDITIKSWRNILRLTQESEACVNGSCFTVMTDGPNSQASYKIFRIEKGKIVKSVMNAPNYHYEE
CSCYPCFSEITCVCRDNWIGNSPPWVFSNONLEYDYGICSGIFGDPNRPNDKTGSCGPVSVSNANGVKGFSFKYGNWGRT
KSISSRNFGEMIWDPNGWTDTDNNSFIKQDINGVINENWSGYSGSFVQHPELTGLDCIRPCFWELIRGRPKENTIWTSGSSISFC
GVNSDTVGWSWPDGA
```

File path /media/data/Workspace/Skripsi-v1.3/data

### Sequence 2

```
MERIKEKDLMSQSRTREILTAKTTVDHMAIIKKYTSGRQEKNIPALRMKWMAMAKYPIADKRIEIMPERNEQGQLWSKTNDA
GSDRVMVSPLAFTWWNRNGPATGTWHYPKVYKTYFEKVERLKHTGFVHFRNOVKIPRRVDINPGHADLSAKEAQDVIMEEVF
PNEVGARTLTSESOLITKEKEELQDKIAPLMVAYMLERELVRKTRFLPVAGGTSSVYIEVHLHTQGTCEQMTPGEVVR
DDVDQSLVIAARNVRRATVSADPQSTQIGGIRMVIDLRONPTEEQAVDICKAAMGLRISSSFSGFTFKRTSGS
SVKKEEEVLGNLDTLKRHVHEGYEEFTMVGRRATAILRKATRRLQDOLIVSGRDEOSIAEAIIIVAMVFSQECMKAVRGDLNF
VNVRANQRNLPMHQLLRHFKQDAKLFQONWGIPEIDNVVMGMIIGLIPDMPTSTEMLRGVRVSKMGVDEYSSTEKVVVSIDRFLRV
RDQGNVLLSPEEVSETQGTEKLYTSSSMWEEINGPESVNLNTYQWIIRNETVKIOWSQDPTMZYKMEFPPSVNKFQKAV
RGOYSGFVRTLFQOMPRDVLGTFDTVOIKLLPFAAAPPEOSMFOSLTVNVRGSGMRILTRGNSPVFNYYKTKRLTVLGKDA
GALTEDPDEGTAGVESAVLRGFLILGKDCKRYGPALSISELSNLAGEKANVLIGQGDVLVMKRKRDSSILTDSQATKRIRM
AIN
```

File path /media/data/Workspace/Skripsi-v1.3/data

### Parameter

Substitution Matrics  Gap Open  Gap Extend

Gambar 4.9 Halaman Home

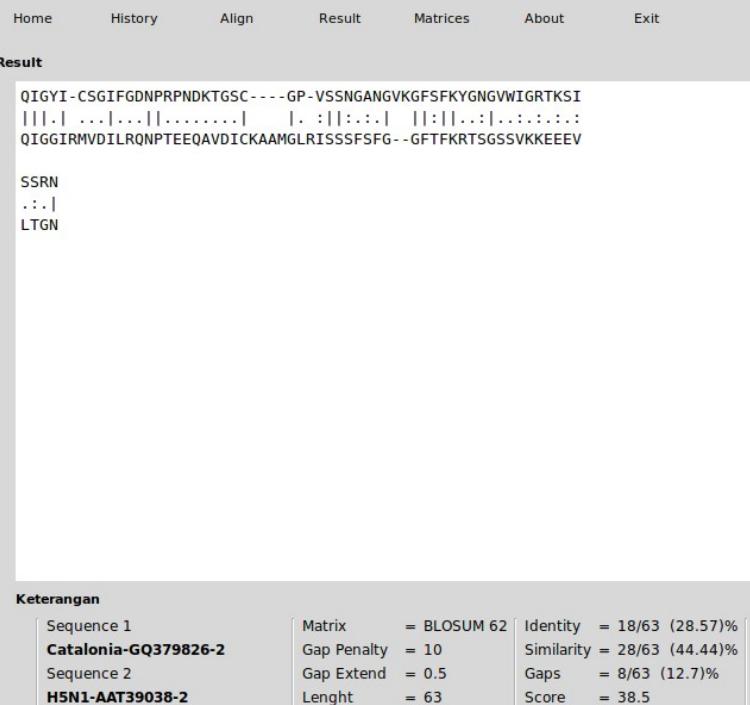
#### Keterangan halaman Home:

1. Data masukan berupa asam amino.
2. *File path* menampilkan alamat *file* yang menjadi data masukan. Jika data dimasukkan secara manual maka *file path* akan tetap kosong.
3. *ComboBox* berisi 3 matriks subtitusi yaitu, BLOSUM45, BLOSUM62, dan BLOSUM80.
4. *SpinBox* *Gap Open* berisi nilai *integer* dengan nilai maksimum 50.
5. *SpinBox* *Gap Extend* berisi nilai *float* dengan nilai maksimum 2.0.
6. Tombol 'Result' dan 'Matrices' tidak aktif ketika tombol 'Align' belum ditekan.

- Tombol 'History' untuk mengambil data masukan terakhir yang telah dimasukkan sebelumnya.
- Tombol 'Reset' untuk mengembalikan parameter ke kondisi awal.

#### 4.3.2 Halaman *Result*

Halaman *Result* menampilkan hasil pensejajaran yang telah dilakukan oleh sistem. Halaman *Result* akan ditampilkan ketika tombol *Align* ditekan. Halaman ini berisi sebuah *TextBox* dan beberapa keterangan tentang hasil pensejajaran. Halaman result secara lengkap ditunjukkan pada gambar 4.10.



Gambar 4.10 Halaman *Result*

### 4.3.3 Halaman Matrices

Halaman *Matrices* berisikan dua buah tabel yang menampilkan hasil perhitungan skor dan penentuan pointer. Elemen tabel tersebut diambil dari nilai tiap elemen yang terdapat pada matriks skor dan matriks pointer. Gambar 4.11 dan 4.12 masing-masing menunjukkan tabel skor dan tabel pointer.

The screenshot shows a web-based application interface for managing matrices. At the top, there is a navigation bar with links: Home, History, Align, Result, Matrices, About, and Exit. Below the navigation bar, the title "Score Matrics" is displayed. The main content area contains a table with 25 rows and 11 columns. The columns are labeled at the top: 339(K), 340(K), 341(E), 342(E), 343(E), 344(V), 345(L), 346(T), 347(G), 348(N), and 349. The rows are labeled on the left with codes such as 241(F), 242(K), 243(Y), etc. The last row, 259(N), has the highest value in the last column, which is highlighted with a blue background. The entire table is enclosed in a scrollable container with vertical and horizontal scroll bars.

	339(K)	340(K)	341(E)	342(E)	343(E)	344(V)	345(L)	346(T)	347(G)	348(N)	349
241(F)	12.5	12	11.5	11	10.5	10	9.5	9	8.5	8	7.5
242(K)	18	17.5	17	16.5	16	15.5	15	14.5	14	13.5	13
243(Y)	16.5	16	15.5	15	14.5	15	14.5	13	12.5	12	12.5
244(G)	16	14.5	14	13.5	13	12.5	12	12.5	19	12.5	10
245(N)	16.5	16	15.5	15	14.5	14	13.5	13	12.5	25	15
246(G)	23	22.5	22	21.5	21	20.5	20	19.5	19	18.5	21
247(V)	21.5	21	20.5	20	19.5	25	21.5	20	17.5	17	19.5
248(W)	27	18.5	18	17.5	17	16.5	23	19.5	18	14.5	15
249(I)	22.5	24	21.5	21	20.5	20	19.5	22	18.5	18	17.5
250(G)	30.5	20.5	22	19.5	19	18.5	18	17.5	28	18.5	17.5
251(R)	24.5	32.5	22.5	22	21.5	21	20.5	20	19.5	28	18.5
252(T)	21	23.5	31.5	21.5	21	21.5	20	25.5	19	19.5	27
253(K)	27.5	26	24.5	32.5	22.5	22	21.5	21	23.5	20	19.5
254(S)	21	27.5	26	24.5	32.5	22.5	22	22.5	21	24.5	20
255(I)	19	21	24.5	23	22.5	35.5	25.5	25	24.5	24	26.5
256(S)	29	20.5	21	24.5	23	25.5	33.5	26.5	25	25.5	22
257(S)	21.5	29	20.5	21	24.5	25	23.5	34.5	26.5	26	23.5
258(R)	30	23.5	29	20.5	21	24.5	23	24.5	32.5	26.5	24
259(N)	25.5	30	23.5	29	20.5	24	22.5	24	24.5	38.5	28.5

Gambar 4.11. Tampilan tabel skor

Gambar di atas menunjukkan tabel skor dengan nilai maksimum serta posisi nilai tersebut. Nilai maksimum tersebut nantinya menjadi posisi awal berjalannya pointer. Pergerakan pointer dapat dilihat pada gambar 4.12.

**Pointer**

	339(K)	340(K)	341(E)	342(E)	343(E)	344(V)	345(L)	346(T)	347(G)	348(N)	349
241(F)	2	2	2	2	2	2	2	2	2	2	2
242(K)	3	3	2	2	2	2	2	2	2	2	2
243(Y)	3	3	3	3	3	3	3	3	2	3	3
244(G)	3	3	3	3	3	2	2	3	3	3	2
245(N)	2	3	2	2	2	2	2	2	3	3	2
246(G)	2	2	2	2	2	2	2	2	3	2	3
247(V)	3	3	3	3	3	3	3	3	2	2	3
248(W)	3	3	3	3	3	3	3	3	3	2	3
249(I)	2	3	2	2	2	3	2	3	2	2	2
<b>250(G)</b>	<b>3</b>	<b>3</b>	<b>3</b>	<b>3</b>	<b>3</b>	<b>2</b>	<b>2</b>	<b>3</b>	<b>3</b>	<b>3</b>	<b>2</b>
<b>251(R)</b>	<b>3</b>	<b>3</b>	<b>2</b>	<b>3</b>	<b>2</b>	<b>2</b>	<b>2</b>	<b>2</b>	<b>2</b>	<b>3</b>	<b>2</b>
<b>252(T)</b>	<b>3</b>	<b>2</b>	<b>3</b>	<b>3</b>							
<b>253(K)</b>	<b>3</b>	<b>3</b>	<b>3</b>	<b>3</b>	<b>3</b>	<b>2</b>	<b>2</b>	<b>2</b>	<b>3</b>	<b>2</b>	<b>2</b>
<b>254(S)</b>	<b>3</b>	<b>3</b>	<b>3</b>	<b>3</b>	<b>3</b>	<b>2</b>	<b>2</b>	<b>3</b>	<b>3</b>	<b>3</b>	<b>2</b>
<b>255(I)</b>	<b>2</b>	<b>1</b>	<b>3</b>	<b>3</b>	<b>1</b>	<b>3</b>	<b>2</b>	<b>2</b>	<b>2</b>	<b>2</b>	<b>3</b>
<b>256(S)</b>	<b>3</b>	<b>1</b>	<b>3</b>	<b>3</b>	<b>3</b>	<b>1</b>	<b>3</b>	<b>3</b>	<b>3</b>	<b>3</b>	<b>3</b>
<b>257(S)</b>	<b>3</b>	<b>3</b>	<b>3</b>	<b>3</b>	<b>3</b>	<b>1</b>	<b>3</b>	<b>3</b>	<b>3</b>	<b>3</b>	<b>3</b>
<b>258(R)</b>	<b>3</b>	<b>3</b>	<b>3</b>	<b>3</b>	<b>3</b>	<b>1</b>	<b>3</b>	<b>1</b>	<b>3</b>	<b>3</b>	<b>3</b>
<b>259(N)</b>	<b>3</b>	<b>3</b>	<b>3</b>	<b>3</b>	<b>3</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>3</b>	<b>3</b>	<b>2</b>
-	-	-	-	-	-	-	-	-	-	-	-

◀ ▶ Nilai maksimum = **38.5** pada posisi Baris = **259** Kolom = **348**

Gambar 4.12. Tampilan tabel pointer

#### 4.4 Implementasi Uji Coba

Uji coba dilakukan dengan beberapa tahap, yaitu :

1. skenario pengujian,
2. evaluasi hasil pengujian, dan
3. analisis hasil pengujian.

Tahapan uji coba yang disebutkan di atas akan dibahas pada subbab selanjutnya.

##### 4.4.1 Skenario pengujian

Pengujian sistem yang dilakukan sesuai dengan rancangan uji coba yang telah dibahas pada subbab 3.5.2 sebelumnya. Pengujian pertama dilakukan pada dua data uji

coba dengan menerapkan nilai *gap penalty* yang berbeda. Pada pengujian ini, parameter penting yang akan dilihat pengaruhnya adalah *gap open* dan *gap extend*. Perubahan nilai skor kesamaan dan skor kemiripan dilihat dari perubahan nilai *gap open* dan *gap extend*. Nilai minimum *gap open* adalah 1 dan nilai maksimum adalah 50, sedangkan nilai minimum *gap extend* adalah 0.1 dan nilai maksimumnya 2.

Pengujian kedua tetap menggunakan data yang sama pada pengujian pertama. Pada pengujian ini, hasil yang diperhatikan adalah pengaruh perubahan nilai *gap open* dan *gap extend* terhadap jumlah kemunculan *gap* pada hasil pensejajaran dan ditinjau dari panjang hasil pensejajaran. Pengujian ketiga dilakukan pada 5 data uji coba yang berbeda dengan nilai *gap open* sama dengan 10 dan *gap extend* sama dengan 0.5. Pengujian dilakukan untuk melihat sebaran *identity score* dan *similarity score* pada masing-masing matriks substitusi.

Data uji coba yang digunakan pada pengujian pertama dan kedua adalah sekuens asam amino [GQ379826]-H1N1 dan [AAT39038]-H5N1 (Lihat lampiran 2). Data uji untuk pengujian ketiga dapat dilihat pada tabel 4.1. Data uji secara terperinci dapat dilihat pada lampiran 3.

Tabel 4.1. Data uji coba

No	Sekuens 1	Sekuens 2
1	[AAR96009] Musa acuminata	[BAG70984] Musa Balbisiana
2	[ACG76157] Phodopus campbelli	[AAF71276] Cricetulus griseus
3	[AAM46197] Panthera leo	[YP_001874850] Panthera pardus
4	[ABM89352] Pongo pygmaeus	[Q2IBE4] Pongo abelii
5	[AAF71276] Cricetulus griseus	[AAF34175] Rattus norvegicus

#### **4.4.2 Evaluasi hasil**

##### **4.4.2.1 Hubungan perubahan nilai *gap penalty* dengan *identity score* dan *similarity score***

Pengujian pertama dilakukan sesuai dengan skenario pengujian yang telah dijelaskan pada bagian 4.4.1, menunjukkan hasil seperti pada tabel 4.2 dan 4.3.



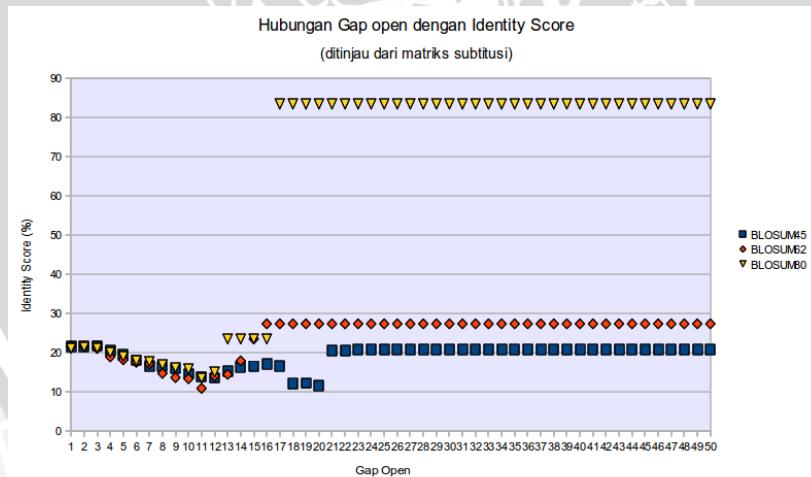
Tabel 4.2. Hasil pengujian *gap open* terhadap *identity* dan *similarity score* berdasarkan matriks substitusi

Gaps	BLOSUM45		BLOSUM62		BLOSUM80	
	Identity	Similarity	Identity	Similarity	Identity	Similarity
1	21.55	29.02	21.53	29.04	21.02	26.56
2	21.47	29.18	21.58	28.42	21.36	26.91
3	21.69	28.17	20.97	26.18	21.13	26.5
4	20.46	26.9	18.83	25.45	20	24.91
5	19.38	26.01	18.08	26.16	18.96	24.4
6	18.01	24.33	17.42	26.71	17.82	25
7	16.52	24.79	16.99	26.7	17.67	25.81
8	16.38	25.44	14.6	24.76	16.8	26.78
9	15.86	24.6	13.54	23.33	16.11	26.16
10	14.5	24.05	13.24	23.53	15.65	25.21
11	13.79	23.42	10.79	18.53	13.43	22.26
12	13.5	24	13.94	23.73	14.93	21.41
13	15.18	27.63	14.37	22.5	23.4	44.68
14	16.28	31.42	17.81	32.88	23.4	42.55
15	16.43	32.16	23.4	44.68	23.4	42.55
16	17.05	32.95	27.27	59.09	23.4	42.55
17	16.52	33.62	27.27	59.09	83.33	100
18	11.96	24.94	27.27	59.09	83.33	100
19	12.09	25.64	27.27	59.09	83.33	100
20	11.54	26.92	27.27	59.09	83.33	100
21	20.45	34.09	27.27	59.09	83.33	100
22	20.45	34.09	27.27	59.09	83.33	100
23	20.69	34.48	27.27	59.09	83.33	100
24	20.69	34.48	27.27	59.09	83.33	100
25	20.69	34.48	27.27	59.09	83.33	100
26	20.69	34.48	27.27	59.09	83.33	100
27	20.69	34.48	27.27	59.09	83.33	100
28	20.69	34.48	27.27	59.09	83.33	100
29	20.69	34.48	27.27	59.09	83.33	100
30	20.69	34.48	27.27	59.09	83.33	100
31	20.69	34.48	27.27	59.09	83.33	100
32	20.69	34.48	27.27	59.09	83.33	100
33	20.69	34.48	27.27	59.09	83.33	100
34	20.69	34.48	27.27	59.09	83.33	100
35	20.69	34.48	27.27	59.09	83.33	100
36	20.69	34.48	27.27	59.09	83.33	100
37	20.69	34.48	27.27	59.09	83.33	100
38	20.69	34.48	27.27	59.09	83.33	100
39	20.69	34.48	27.27	59.09	83.33	100
40	20.69	34.48	27.27	59.09	83.33	100
41	20.69	34.48	27.27	59.09	83.33	100
42	20.69	34.48	27.27	59.09	83.33	100
43	20.69	34.48	27.27	59.09	83.33	100
44	20.69	34.48	27.27	59.09	83.33	100
45	20.69	34.48	27.27	59.09	83.33	100
46	20.69	34.48	27.27	59.09	83.33	100
47	20.69	34.48	27.27	59.09	83.33	100
48	20.69	34.48	27.27	59.09	83.33	100
49	20.69	34.48	27.27	59.09	83.33	100
50	20.69	34.48	27.27	59.09	83.33	100

Tabel 4.3. Hasil pengujian *gap extend* terhadap *identity* dan *similarity score* berdasarkan matriks substitusi

Gap	BLOSUM45		BLOSUM62		BLOSUM80	
	Identity	Similarity	Identity	Similarity	Identity	Similarity
0.1	21.55	29.02	21.53	29.04	21.02	26.56
0.2	21.9	29.32	21.46	29.25	21.76	27.38
0.3	22.54	29.79	22.22	29.8	22.39	28.83
0.4	22.58	30.29	22.37	30	22.45	28.9
0.5	22.69	30.61	22.62	30.03	22.54	28.89
0.6	22.72	30.65	22.27	30.8	22.79	29.12
0.7	22.72	30.65	23	32.12	23.27	29.52
0.8	23.36	32.17	22.93	32.49	23.13	29.66
0.9	23.24	32.68	22.93	32.49	23.13	29.66
1	22.87	33.1	22.88	32.37	23.01	30.68
1.1	22.87	33.1	22.88	32.37	23.01	30.68
1.2	22.87	33.1	22.88	32.37	23.01	30.68
1.3	22.87	33.1	22.88	32.37	23.01	30.68
1.4	22.87	33.1	22.88	32.37	23.01	30.68
1.5	22.87	33.1	22.88	32.37	23.01	30.68
1.6	22.87	33.1	22.88	32.37	23.01	30.68
1.7	22.87	33.1	22.88	32.37	23.01	30.68
1.8	22.87	33.1	22.88	32.37	23.01	30.68
1.9	22.87	33.1	22.88	32.37	23.01	30.68
2	22.87	33.1	22.88	32.37	23.01	30.68

Berdasarkan tabel 4.2 didapatkan sebuah grafik fungsi yang ditunjukkan pada gambar 4.13 dan 4.14 sebagai berikut.

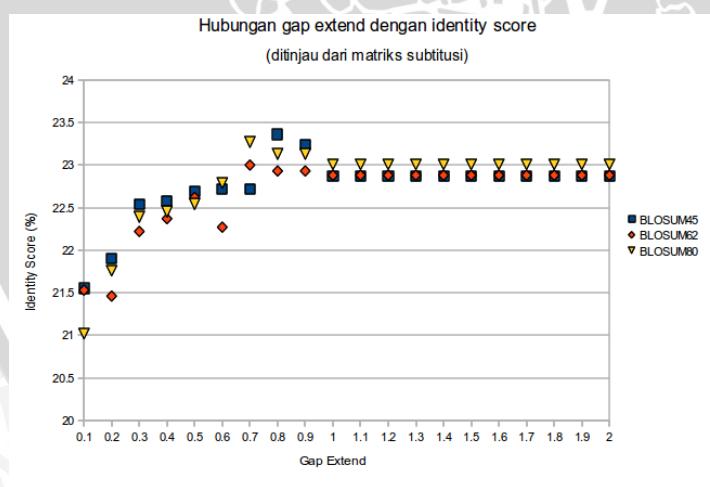


Gambar 4.13 Grafik hubungan *gap open* dengan *identity score*

Gambar 4.13 menunjukkan hubungan antara perubahan nilai *gap open* dengan *identity score*. Titik-titik adalah representasi nilai dari *identity score* pada masing-masing matriks substitusi.

*Identity score* dengan menggunakan BLOSUM45 mencapai kondisi konstan pada (*gap open* = 23). *Identity score* dengan BLOSUM62 konstan pada (*gap open* = 16), dan untuk BLOSUM80 konstan pada (*gap open* = 17).

Grafik hubungan antara *gap open* dengan *identity score* menunjukkan sebuah kondisi tertentu. Kondisi tersebut adalah nilai *identity score* mengalami penurunan sampai pada titik minimum. Pada saat *identity score* mencapai titik minimum, nilai tersebut akan mengalami kenaikan dan mengalami kondisi konstan. Hal tersebut disebabkan karena hasil pencejajaran yang didapatkan masih memiliki jumlah gap yang sangat besar (lihat tabel 4.4). Kondisi konstan pada gambar 4.13 juga disebabkan oleh hasil pencejajaran yang tidak memiliki gap sama sekali, sehingga *gap open* tidak berpengaruh lagi terhadap perubahan *identity score*. Contohnya pada *identity score* ditinjau dari BLOSUM45, nilai *identity score* tidak mengalami perubahan pada *gap open* > 23.

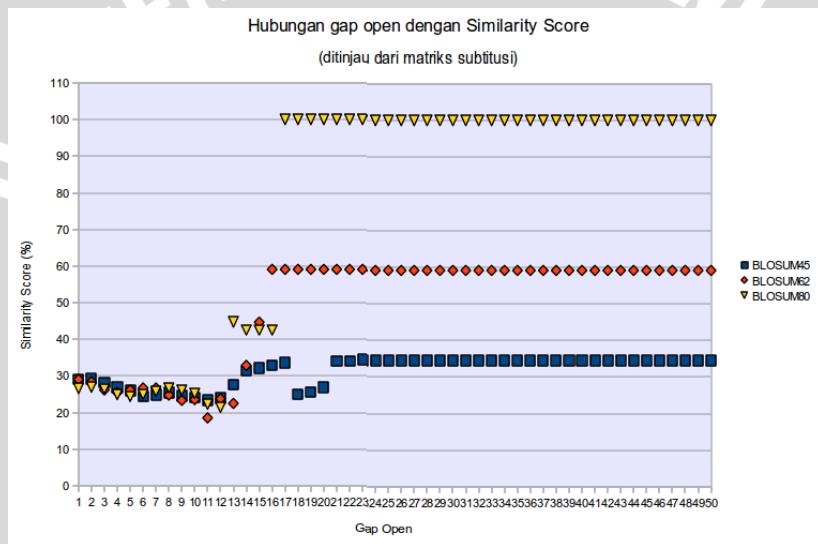


Gambar 4.14. Grafik hubungan *gap extend* dengan *identity score*

Gambar 4.14 menunjukkan grafik hubungan gap extend dengan *identity score* ditinjau dari matriks substitusi. Titik-titik pada grafik merepresentasikan nilai *identity score* yang dipengaruhi oleh variabel independen yaitu *gap extend*.

Titik konsisten dimana *identity score* tetap sama pada kondisi nilai *gap extend* tetap bertambah terletak pada nilai *gap extend* sama dengan 1 (satu). Hal tersebut terjadi pada semua BLOSUM.

Berdasarkan tabel 4.3 didapatkan sebuah grafik fungsi yang ditunjukkan pada gambar 4.15 dan 4.16 sebagai berikut.



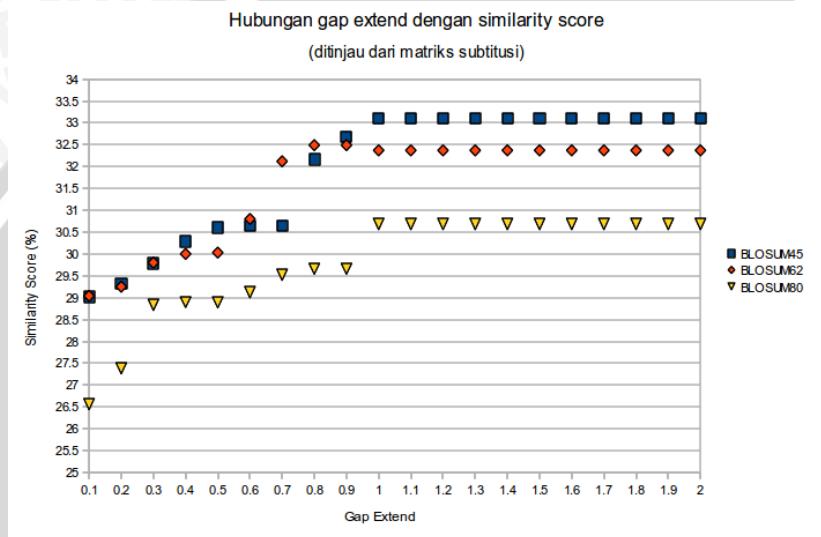
Gambar 4.15. Grafik hubungan gap open dengan *similarity score*

Gambar 4.15 menunjukkan hubungan antara perubahan nilai gap open dengan *similarity score*. Perubahan *similarity score* yang diakibatkan oleh perubahan nilai *gap open* ditunjukkan oleh titik-titik pada grafik.

*Similarity score* dengan menggunakan BLOSUM45 mencapai kondisi konstan pada (*gap open* = 21). *Similarity score* dengan BLOSUM62 konstan pada (*gap open* = 16), dan untuk BLOSUM80 konstan pada (*gap open* = 17).

Gambar 4.15 menunjukkan suatu kondisi yang sama dengan kondisi yang ditunjukkan oleh gambar 4.13. Kondisi

tersebut juga disebabkan oleh jumlah *gap* pada hasil pensejajaran pada awalnya memiliki nilai yang sangat besar sehingga hasil perhitungan *similarity score* semakin menurun (lihat persamaan 3.2). Kondisi konstan dicapai ketika hasil pensejajaran tidak memiliki gap.



Gambar 4.16. Hubungan gap extend dengan *similarity score*

Gambar 4.16 menunjukkan hubungan antara perubahan nilai *gap extend* dengan *similarity score*. Titik-titik yang ditunjukkan pada gambar 4.16 adalah representasi dari *similarity score* pada masing-masing matriks subtitusi.

Titik konsisten dimana *identity score* tetap sama pada kondisi nilai *gap extend* tetap bertambah terletak pada nilai *gap extend* sama dengan 1 (satu). Hal tersebut terjadi pada semua BLOSUM.

#### 4.4.2.2 Hubungan perubahan *gap penalty* terhadap jumlah kemunculan *gap* pada hasil pensejajaran

Pengujian kedua lebih difokuskan pada pengaruh perubahan nilai *gap open* dan *gap extend* terhadap jumlah kemunculan *gap* pada hasil pensejajaran. Hasil pengujian ini dapat

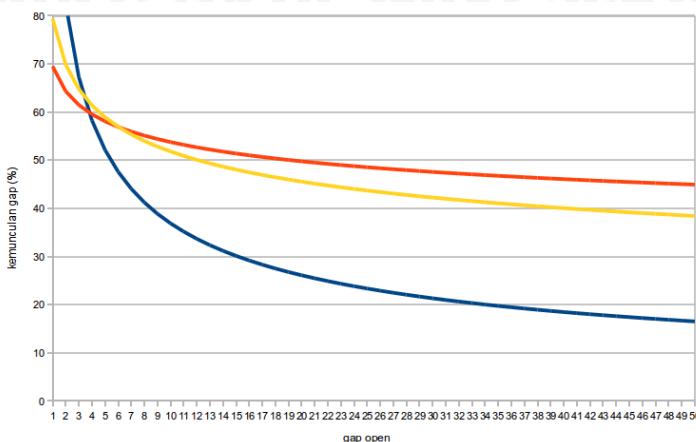
dilihat pada tabel 4.4 dan 4.5.

Tabel 4.4. Hasil pengujian terhadap pengaruh *gap open* pada kemunculan *gap* pada hasil pensejajaran

Gap Open	BLOSUM45		BLOSUM62		BLOSUM80	
	Gaps	length	Gaps	length	Gaps	length
1	66.29	789	65.61	785	69.54	847
2	64.4	778	63.7	774	67.65	810
3	61.9	756	65.63	806	66	800
4	62.03	777	64.89	786	65.16	795
5	61.9	769	59.45	730	65.74	791
6	62.1	744	55.3	689	62.29	724
7	57.98	702	50.97	618	58.53	651
8	53.58	629	53.81	630	54.04	631
9	52.75	618	54	613	53.11	627
10	52.55	607	53.76	612	52.55	607
11	51.33	602	63.31	556	58.48	566
12	50.83	600	46.33	531	60.28	355
13	44.55	514	58.13	320	42.55	47
14	33.94	436	56.16	73	42.55	47
15	30.05	426	42.55	47	42.55	47
16	24.72	352	0	22	42.55	47
17	24.22	351	0	22	0	6
18	34.86	393	0	22	0	6
19	30.04	273	0	22	0	6
20	34.62	182	0	22	0	6
21	7.95	88	0	22	0	6
22	7.95	88	0	22	0	6
23	0	58	0	22	0	6
24	0	58	0	22	0	6
25	0	58	0	22	0	6
26	0	58	0	22	0	6
27	0	58	0	22	0	6
28	0	58	0	22	0	6
29	0	58	0	22	0	6
30	0	58	0	22	0	6
31	0	58	0	22	0	6
32	0	58	0	22	0	6
33	0	58	0	22	0	6
34	0	58	0	22	0	6
35	0	58	0	22	0	6
36	0	58	0	22	0	6
37	0	58	0	22	0	6
38	0	58	0	22	0	6
39	0	58	0	22	0	6
40	0	58	0	22	0	6
41	0	58	0	22	0	6
42	0	58	0	22	0	6
43	0	58	0	22	0	6
44	0	58	0	22	0	6
45	0	58	0	22	0	6
46	0	58	0	22	0	6
47	0	58	0	22	0	6
48	0	58	0	22	0	6
49	0	58	0	22	0	6
50	0	58	0	22	0	6

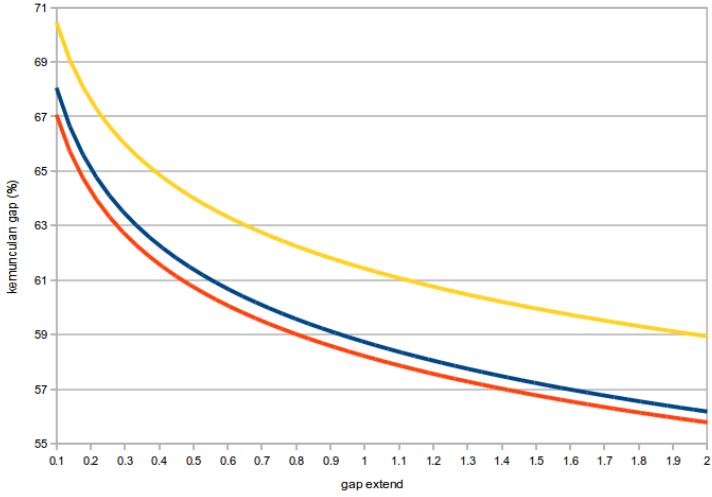
Tabel 4.5. Hasil pengujian terhadap pengaruh *gap extend* pada kemunculan *gap* pada hasil pensejajaran

Gaps	BLOSUM45		BLOSUM62		BLOSUM80	
	Gaps	length	Gaps	length	Gaps	length
0.1	66.29	789	65.61	785	69.54	847
0.2	65.69	781	65.26	783	68.09	818
0.3	64.77	772	63.53	765	65.64	777
0.4	63.71	766	62.63	760	65.29	775
0.5	62.27	758	61.9	756	64.77	772
0.6	62.09	757	60.8	750	64.43	759
0.7	59.89	718	58.91	713	63.13	735
0.8	59.3	715	58.51	711	63.13	735
0.9	58.31	710	58.51	711	63.13	735
1	57.1	704	56.69	695	59.69	717
1.1	57.1	704	56.69	695	59.69	717
1.2	57.1	704	56.69	695	59.69	717
1.3	57.1	704	56.69	695	59.69	717
1.4	57.1	704	56.69	695	59.69	717
1.5	57.1	704	56.69	695	59.69	717
1.6	57.1	704	56.69	695	59.69	717
1.7	57.1	704	56.69	695	59.69	717
1.8	57.1	704	56.69	695	59.69	717
1.9	57.1	704	56.69	695	59.69	717
2	57.1	704	56.69	695	59.69	717



Gambar 4.17. Grafik hubungan antara *gap open* dengan kemunculan *gap*

Gambar 4.17 menunjukkan sebuah bentuk hubungan antara perubahan *gap open* terhadap nilai kemunculan *gap* pada hasil pensejajaran. Kemunculan *gap* semakin berkurang jika nilai *gap open* semakin besar. Warna biru mewakili BLOSUM45, warna merah mewakili BLOSUM62, dan warna kuning menunjukkan BLOSUM80. Grafik di atas menunjukkan aproksimasi dari kemunculan *gap* pada hasil pensejajaran yang ditinjau dari perubahan nilai *gap open*. Hal tersebut dapat dilihat pada BLOSUM45 yang memiliki hasil pensejajaran tanpa *gap* pada *gap open* = 23. BLOSUM62 memiliki hasil tanpa *gap* pada *gap open* = 16 dan BLOSUM80 pada *gap open* = 17 (Lihat tabel 4.4).



Gambar 4.18. Grafik hubungan antara *gap extend* dengan kemunculan *gap*

Gambar 4.18 menunjukkan sebuah bentuk hubungan antara perubahan *gap extend* terhadap nilai kemunculan *gap* pada hasil pensejajaran. Kemunculan *gap* semakin berkurang jika nilai *gap extend* semakin besar. Persentase kemunculan *gap* akan mencapai kondisi konstan pada *gap* yang besar. Jika dilihat pada gambar 4.18, maka dapat diketahui bahwa semua matriks substitusi menunjukkan hasil yang sama. Yaitu, semua matriks substitusi memiliki kemunculan *gap* konstan pada *gap extend* = 1 (lihat tabel 4.5).

#### 4.4.2.3 Hubungan matriks substitusi terhadap *identity score* dan *similarity score*

Pengujian ketiga yang dilakukan pada 5 buah data uji coba ditunjukkan pada tabel 4.6 dan 4.7 berikut.

Tabel 4.6. Tabel hubungan matriks substitusi dengan *identity score*

No	Data Uji Coba	<i>identity score (%)</i>		
		BLOSUM45	BLOSUM62	BLOSUM80
1	[AAR96009] & [BAG70984]	17,34	20,24	21,43
2	[ACG76157] & [AAF71276]	19,86	21,22	29,9
3	[AAM46197] & [YP_001874850]	97,52	97,52	97,52
4	[ABM89352] & [Q2IBE4]	19,45	19,3	25
5	[AAF71276] & [AAF34175]	17,77	17,53	41,67

Tabel 4.7. Tabel hubungan matriks substitusi dengan *identity score*

No	Data Uji Coba	<i>similarity score (%)</i>		
		BLOSUM45	BLOSUM62	BLOSUM80
1	[AAR96009] & [BAG70984]	32,52	35,71	35
2	[ACG76157] & [AAF71276]	32,71	33,76	41,24
3	[AAM46197] & [YP_001874850]	99,17	99,01	99,01
4	[ABM89352] & [Q2IBE4]	34,3	36,26	36,46
5	[AAF71276] & [AAF34175]	29,86	34,25	54,17

#### 4.4.3 Validasi Hasil

Validasi dilakukan dengan membandingkan hasil pensejajaran oleh sistem dengan hasil pensejajaran pada sistem EMBOSS. Skenario dibahas pada subbab 3.5.3. Hasil pensejajaran dengan menggunakan sistem EMBOSS ditunjukkan pada gambar 4.19.

QIGYI-CSGIFGDNPBPNDKTGSCGP----VSSNGANGVKGFSFKYGN  
| | . . . | . . . | . . . . | .. : | : . . | || : | : . . |  
QIGGIRMVDILRQNPTEEQAVDICKAAMGLRISSSFSG--GFTFKRTSG

VWIGRTKSISSRN  
.: . . . . . . |  
SSVKKEEVLTGN

Gambar 4.19. Hasil pensejajaran dengan menggunakan EMBOSS

Pensejajaran dengan menggunakan sistem yang dibuat memberikan hasil yang ditunjukkan pada gambar 4.20.

QIGYI-CSGIFGDNPBPNDKTGSC---GP-VSSNGANGVKGFSFKYGNVWIGRTKSI  
| | . . . | . . . | . . . . | .. : | : . . | || : | . . . . . . . . .  
QIGGIRMVDILRQNPTEEQAVDICKAAMGLRISSSFSG--GFTFKRTSGSSVKKEEEV

SSRN  
. . . |  
LTGN

Gambar 4.20. Hasil pensejajaran dengan sistem yang dibuat

Hasil pensejajaran menggunakan sistem EMBOSS dengan sistem yang dibuat memiliki perbedaan. Perbedaan tersebut dapat ditunjukkan pada nilai hasil pensejajaran (ditunjukkan pada tabel 4.8).

Tabel 4.8. Perbandingan hasil pensejajaran

Parameter penelitian	Sistem yang dibuat	Sistem EMBOSS
Identity score	28,57	27,0
Similarity score	44,44	42,9
Panjang hasil	63	63
Kemunculan gap	12,7	12,7
Skor total	38,5	38,5

Nilai pensejajaran dengan menggunakan sistem EMBOSS

menunjukkan beberapa kesamaan dengan sistem yang dibuat. Panjang hasil pensejajaran skor total menunjukkan nilai yang sama yaitu 63 dan 38,5. *Identity score* pada hasil pensejajaran dengan sistem EMBOSS lebih kecil dari sistem yang dibuat yaitu 27% dan 28,57%. *Similarity score* pada sistem EMBOSS juga lebih kecil dari sistem yang dibuat, yaitu 42,9% dan 44,44%.

#### 4.4.4 Analisis Hasil

##### 4.4.4.1 Hubungan perubahan nilai *gap penalty* dengan *identity score* dan *similarity score*

Nilai *gap penalty* merupakan variabel independen yang mempengaruhi besar kecilnya *identity score* dan *similarity score*. Gambar 4.13 dan 4.14 masing-masing menunjukkan grafik fungsi pengaruh *gap open* terhadap *identity score* dan *similarity score*, dimana semakin besar nilai *gap open* maka semakin besar pula *identity score* dan *similarity score*. Namun, pada suatu kondisi tertentu *identity score* dan *similarity score* akan mencapai nilai konstan. Konsistensi nilai tersebut dapat dilihat pada saat *gap penalty* mencapai nilai terbesar tertentu. Berdasarkan nilai konstan pada grafik fungsi, BLOSUM80 dapat menghasilkan pensejajaran dengan *identity score* dan *similarity score* yang tinggi dibandingkan dengan matriks substitusi lainnya ditinjau dari besar *gap penalty*.

Gambar 4.15 dan 4.16 masing-masing menunjukkan grafik fungsi pengaruh atau hubungan antara *gap extend* dengan *identity score* dan *similarity score*. Grafik fungsi tersebut menunjukkan bahwa bahwa semakin besar nilai *gap extend* yang diterapkan pada BLOSUM45 dan BLOSUM80, maka akan semakin besar pula *identity* dan *similarity score*-nya.

##### 4.4.4.2 Hubungan perubahan *gap penalty* terhadap jumlah kemunculan gap pada hasil pensejajaran

*Gap penalty* menentukan banyaknya kemunculan gap pada hasil pensejajaran. Hubungan kausal tersebut dapat dilihat pada gambar 4.17 dan 4.18. Grafik 4.17 menunjukkan

bahwa koefisien regresi terhadap kemunculan gap terjadi penurunan. Artinya, semakin besar nilai gap open maka akan semakin kecil kemungkinan munculnya gap pada hasil pensejajaran. Demikian juga pada gap extend, semakin besar nilai gap extend maka kemunculan gap pada hasil pensejajaran akan semakin kecil. Hal tersebut dapat dilihat pada nilai koefisien regresi yang bernilai negatif.

#### **4.4.4.3 Hubungan matriks substitusi terhadap *identity score* dan *similarity score***

Matriks substitusi menentukan tingkat mutasi yang dialami oleh asam amino. Tingkat mutasi tersebut mempengaruhi kemungkinan kemiripan suatu pasangan sekuens. Pengaruh yang dapat dilihat adalah pengaruhnya pada *identity score* dan *similarity score* dari hasil pensejajaran. Pada tabel 4.6 dan 4.7, ditunjukkan bahwa semakin tinggi tingkat cluster matriks substitusi maka semakin tinggi juga *identity score* dan *similarity score* hasil pensejajaran. Contohnya, data [ACG76157] &[AAF71276] menunjukkan bahwa BLOSUM80 memberikan hasil yang lebih tinggi dibandingkan matriks BLOSUM lainnya.

#### **4.4.4.4 Analisis validasi hasil**

Hasil pensejajaran yang telah tervalidasi menunjukkan perbedaan yang tidak signifikan. Berdasarkan tabel 4.8 menunjukkan perbedaan pada *identity score* dan *similarity score*. Hasil pensejajaran dengan menggunakan sistem yang dibuat menunjukkan nilai identity dan similarity score lebih besar dibandingkan hasil pensejajaran oleh sistem EMBOSS. Hal tersebut disebabkan oleh pendekatan yang dilakukan oleh sistem dalam mencari kesamaan sekuens.

Sistem EMBOSS mengabaikan asam amino G pada sekuens kedua sehingga asam amino G pada sekuens pertama tidak bergeser untuk disejajarkan dengan asam amino G pada sekuens kedua (ditunjukkan oleh kotak merah pada gambar 4.21).

QIGYI-CSGIFGDNPRNDKTGCGP----VSSNGANGVKGFSFKYGN  
|||.| ...|...| |.....|... :| :...| ||:|...:  
QIGGIRMVDILRQNPTEEQAVDICKAAMGLRISSSFSFG--GFTFKRTSG  
  
VWIGRTKSISSRN  
...:...:...:...|  
SSVKKEEVLTGN

Gambar 4.21. Analisis hasil pencejajaran dengan sistem EMBOSS

Sistem yang dibuat melakukan pencejajaran dengan mensejajarkan asam amino G sekuens pertama dengan asam amino G pada sekuens kedua (ditunjukkan oleh kotak merah pada gambar 4.22). Hal tersebut menunjukkan sistem yang telah dibuat melakukan pencejajaran dengan mencari semua asam amino yang sejajar pada sekuens yang disejajarkan.

QIGYI-CSGIFGDNPRNDKTGSC---GP-VSSNGANGVKGFSFKYGNVWIGRTKSI  
|||.| ...|...| |.....|... :| :...| ||:|...:|...:...:  
QIGGIRMVDILRQNPTEEQAVDICKAAMGLRISSSFSFG--GFTFKRTSGSSVKKEEV  
  
SSRN  
...|  
LTGN

Gambar 4.22. Analisis hasil pencejajaran dengan sistem yang dibuat

UNIVERSITAS BRAWIJAYA



## BAB 5

# PENUTUP

### 5.1 Kesimpulan

Berdasarkan hasil dan pembahasan sebelumnya, maka dapat disimpulkan.

1. Metode Smith-Waterman dapat diterapkan dalam melakukan pencejajaran sekuen asam amino dengan mengikuti kaidah substitusi tingkat mutasi asam amino.
2. Gap penalty menentukan besar kecilnya nilai identity score dan similarity score. Nilai identity score dan similarity score akan mencapai kondisi ekuilibrium terhadap nilai gap open. Kondisi tersebut disebabkan oleh jumlah gap pada hasil pencejajaran.
3. Semakin besar tinggi tingkat kluster sebuah matriks substitusi maka semakin besar pula identity score dan similarity score. Jadi, BLOSUM80 menghasilkan nilai identity dan similarity score yang lebih besar dibandingkan dengan BLOSUM45 dan BLOSUM62.

### 5.2 Saran

Penelitian ini merupakan sebuah penelitian awal sehingga penulis menyarankan beberapa hal sebagai berikut.

1. Untuk melakukan penelitian lanjutan maka dapat dicoba dengan membandingkan metode Smith-Waterman dengan metode Needleman-Wunsch dalam proses pencejajaran sekuen asam amino.
2. Penelitian ini masih menggunakan *pairwise alignment*, sehingga penelitian yang lebih lanjut dapat menerapkan *multiple alignment*.

UNIVERSITAS BRAWIJAYA



## DAFTAR PUSTAKA

- Altschul, SF. 1997. *Gapped BLAST and PSI-BLAST: a new generation of protein database search programs*. Nucleic Acids Research, Vol 25, Issue 17 3389-3402.  
<http://nar.oxfordjournals.org/cgi/content/abstract/25/17/>  
tanggal akses : 19 Mei 2009
- EBI Team. 2008. About Gap in Sequence Alignments.  
<http://www.ebi.ac.uk/help/gaps.html>, Tanggal akses : 26 Juni 2009
- Eddy, Sean R. 2004. **BLOSUM62 Alignment Score Matrix**. Nature Biotechnology volume 22.  
<http://www.nature.com/naturebiotechnology> tanggal akses 20 April 2009
- Farrar, Michael. 2006. *Striped Smith-Waterman speeds database searches six times over other SIMD implementations*. November 16, 2006 Bioinformatics 2007 23(2):156-161; doi:10.1093/bioinformatics/btl582  
<http://bioinformatics.oxfordjournals.org/cgi/content/> tanggal akses : 20 Mei 2009
- Gibas, Cynthia dan Per Jambeck. 2001. *Developing Bioinformatics Computer Skills*. O'Reilly. USA
- Henikoff, S dan J.G. Henikoff. 1992. *Amino acid substitution matrices from protein blocks*. PNAS November 15, 1992 vol. 89 no. 22 10915-10919.  
<http://www.pnas.org/content/89/22/10915.abstract> tanggal akses : 20 Mei 2009
- Lesk, Arthur M. 2002. *Introductions to Bioinformatics*. Oxford University Press. New York

Nicholas, Hugh. 2000. *A Tutorial on Searching Sequence Databases and Sequence Scoring Methods*. Journal Biotechniques. Vol.28, No. 6, June 2000  
<http://bioinformatics.albany.edu/scoring.htm> tanggal akses : 7 Mei 2009

Nugroho, Adi. 2008. *Algoritma dan Struktur Data dalam Bahasa Java*. Andi Jogja. Jogjakarta.

Pearson, WR dan DJ Lipman. 1988. *Improved tools for biological sequence comparison*. PNAS April 1, 1988 vol. 85 no. 8 2444-2448. <http://www.pnas.org/content/85/8/2444.abstract> tanggal akses : 20 Mei 2009

Polanski, Andrzej dan Marek Kimmel. 2007. *Bioinformatics*. Springer. Berlin

Smith, TF dan Waterman MS. 1981. *Identification of Common Molecular Subsequences*. Journal Molecular Biology. Chapter 147.

Suryo. 2001. **Genetika**. Gadjah Mada Press. Jogjakarta

Wernersson,R dan Pedersen AG. 2003. *RevTrans : Multiple Alignment of Coding DNA From Aligned Amino Acid Sequences*.  
<http://nar.oxfordjournals.org/cgi/content/full/31/13/3537>  
tanggal akses : 4 Januari 2009

Xing, Yun. *Sequence Alignment Algorithms*.  
<http://icl.pku.edu.cn/yujs/papers> tanggal akses : 20 Maret 2009

Yuwono, Triwibowo. 2002. *Biologi Molekuler*. Erlangga. Jakarta

## Lampiran 1 Hasil perhitungan manual

Baris ke-1

S[1,1]= maks (0, 4, -0.5, -0.5)

S[1,1]= 4

P[1,1]= 0

S[1,2]= maks (0, -3, -0.5, -0.5)

S[1,2]= 0

P[1,2]= 0

S[1,3]= maks (0, -3, -0.5, -0.5)

S[1,3]= 0

P[1,3]= 0

S[1,4]= maks (0, 0, -0.5, -0.5)

S[1,4]= 0

P[1,4]= 0

S[1,5]= maks (0, 4, -0.5, -0.5)

S[1,5]= 4

P[1,5]= 0

S[1,6]= maks (0, 0, -0.5, -0.5)

S[1,6]= 0

P[1,6]= 0

S[1,7]= maks (0, 0, -0.5, -0.5)

S[1,7]= 0

P[1,7]= 0

S[1,8]= maks (0, 4, -0.5, -0.5)

S[1,8]= 4

P[1,8]= 0

Baris ke-2

S[2,1]= maks (0, -3, 3, 3)

S[2,1]= 3

P[2,1]= 0

S[2,2]= maks (0, 10, -1, -1)

S[2,2]= 10

P[2,2]= 0

S[2,3]= maks (0, -1, -1, -1)

S[2,3]= 0

P[2,3]= 0

S[2,4]= maks (0, -2, -1, -1)

S[2,4]= 0

P[2,4]= 0

S[2,5]= maks (0, -3, 3, 3)

S[2,5]= 3

P[2,5]= 0

S[2,6]= maks (0, 2, -1, -1)

S[2,6]= 2

P[2,6]= 0

S[2,7]= maks (0, -2, -1, -1)

S[2,7]= 0

P[2,7]= 0

S[2,8]= maks (0, -3, 3, 3)

S[2,8]= 3

P[2,8]= 0

Baris ke-3

S[3,1]= maks (0, -3, 2.5, 2.5)

S[3,1]= 2.5

P[3,1]= 0

S[3,2]= maks (0, 2, 9, 9)

S[3,2]= 9

P[3,2]= 0

S[3,3]= maks (0, 18, -1, -1)

S[3,3]= 18

P[3,3]= 0

S[3,4]= maks (0, -2, -1, -1)

S[3,4]= 0

P[3,4]= 0

S[3,5]= maks (0, -3, 2.5, 2.5)

S[3,5]= 2.5

P[3,5]= 0

S[3,6]= maks (0, 1, 1, 1)

S[3,6]= 1

P[3,6]= 0

S[3,7]= maks (0, 0, -1, -1)

S[3,7]= 0

P[3,7]= 0

S[3,8]= maks (0, -3, 2.5, 2.5)

S[3,8]= 2.5

P[3,8]= 0

Baris ke-4

S[4,1]= maks (0, 0, 2, 2)

S[4,1]= 2  
P[4,1]= 0  
S[4,2]= maks (0, 0.5, 8.5, 8.5)  
S[4,2]= 8.5  
P[4,2]= 0  
S[4,3]= maks (0, 7, 17, 17)  
S[4,3]= 17  
P[4,3]= 0  
S[4,4]= maks (0, 22, -1, -1)  
S[4,4]= 22  
P[4,4]= 0  
S[4,5]= maks (0, 0, 2, 2)  
S[4,5]= 2  
P[4,5]= 0  
S[4,6]= maks (0, 6.5, 0.5, 0.5)  
S[4,6]= 6.5  
P[4,6]= 0  
S[4,7]= maks (0, 5, -1, -1)  
S[4,7]= 5  
P[4,7]= 0  
S[4,8]= maks (0, 0, 2, 2)  
S[4,8]= 2  
P[4,8]= 0

Baris ke-5  
S[5,1]= maks (0, 3, 1.5, 1.5)  
S[5,1]= 3  
P[5,1]= 0  
S[5,2]= maks (0, -1, 8, 8)  
S[5,2]= 8  
P[5,2]= 0  
S[5,3]= maks (0, 5.5, 16.5, 16.5)  
S[5,3]= 16.5  
P[5,3]= 0  
S[5,4]= maks (0, 16, 21, 21)  
S[5,4]= 21  
P[5,4]= 0  
S[5,5]= maks (0, 25, 1.5, 1.5)  
S[5,5]= 25  
P[5,5]= 0  
S[5,6]= maks (0, 1, 5.5, 5.5)  
S[5,6]= 5.5

P[5,6]= 0  
S[5,7]= maks (0, 5.5, 4, 4)  
S[5,7]= 5.5  
P[5,7]= 0  
S[5,8]= maks (0, 8, 1.5, 1.5)  
S[5,8]= 8  
P[5,8]= 0  
  
Baris ke-6  
S[6,1]= maks (0, 4, 2, 2)  
S[6,1]= 4  
P[6,1]= 0  
S[6,2]= maks (0, 0, 7.5, 7.5)  
S[6,2]= 7.5  
P[6,2]= 0  
S[6,3]= maks (0, 5, 16, 16)  
S[6,3]= 16  
P[6,3]= 0  
S[6,4]= maks (0, 16.5, 20.5, 20.5)  
S[6,4]= 20.5  
P[6,4]= 0  
S[6,5]= maks (0, 25, 24, 24)  
S[6,5]= 25  
P[6,5]= 0  
S[6,6]= maks (0, 25, 5, 5)  
S[6,6]= 25  
P[6,6]= 0  
S[6,7]= maks (0, 5.5, 4.5, 4.5)  
S[6,7]= 5.5  
P[6,7]= 0  
S[6,8]= maks (0, 9.5, 7, 7)  
S[6,8]= 9.5  
P[6,8]= 0  
  
Baris ke-7  
S[7,1]= maks (0, 3, 3, 3)  
S[7,1]= 3  
P[7,1]= 0  
S[7,2]= maks (0, 1, 7, 7)  
S[7,2]= 7  
P[7,2]= 0  
S[7,3]= maks (0, 4.5, 15.5, 15.5)

S[7,3]= 15.5  
P[7,3]= 0  
S[7,4]= maks (0, 15, 20, 20)  
S[7,4]= 20  
P[7,4]= 0  
S[7,5]= maks (0, 23.5, 24, 24)  
S[7,5]= 24  
P[7,5]= 0.0  
S[7,6]= maks (0, 24, 24, 24)  
S[7,6]= 24  
P[7,6]= 0  
S[7,7]= maks (0, 24, 4.5, 4.5)  
S[7,7]= 24  
P[7,7]= 0  
S[7,8]= maks (0, 8.5, 8.5, 8.5)  
S[7,8]= 8.5  
P[7,8]= 0

Baris ke-8  
S[8,1]= maks (0, 4, 2.5, 2.5)  
S[8,1]= 4  
P[8,1]= 0

S[8,2]= maks (0, 0, 6.5, 6.5)  
S[8,2]= 6.5  
P[8,2]= 0  
S[8,3]= maks (0, 4, 15, 15)  
S[8,3]= 15  
P[8,3]= 0  
S[8,4]= maks (0, 15.5, 19.5, 19.5)  
S[8,4]= 19.5  
P[8,4]= 0  
S[8,5]= maks (0, 24, 23.5, 23.5)  
S[8,5]= 24  
P[8,5]= 0  
S[8,6]= maks (0, 24, 23.5, 23.5)  
S[8,6]= 24  
P[8,6]= 0  
S[8,7]= maks (0, 24, 23, 23)  
S[8,7]= 24  
P[8,7]= 0  
S[8,8]= maks (0, 28, 8, 8)  
S[8,8]= 28  
P[8,8]= 0

## Lampiran 2 Data uji coba untuk pengujian pertama dan kedua

### Influenza A virus (A/Catalonia/405/2009(H1N1)) segment 6 neuraminidase (NA) gene, partial cds

LOCUS G0379826 1230 bp cRNA linear VRL 15-JUL-2009  
DEFINITION Influenza A virus (A/Catalonia/405/2009(H1N1)) segment 6 neuraminidase (NA) gene, partial cds.

#### ORIGIN

```
1 ggcgtcattac ttatggaaac aacacttggg taaatcagac atatgttaac atcagcaaca
61 ccacatttgc tgctggacag tcagtgttt ccgtaaaatt agccggcaat tccttcctct
121 gcccgttag tgatgggcata atatacaga aagacaacat tgtaagaatc ggttccaagg
181 ggatgtgtc tgctataagg gaaaccattcatacatgtcc ccccttggaa tgccagaaat
241 tcttcgttag tcacaggggccatc tgcttataatg aacaaccatc caatggaaacc attaaagaca
301 ggagccccata tcgaaccctta atgagctgtc ctatggtga agttccctct ccataacaact
361 caatggatc tgcaatggggcata tgatggcata tgatggcata tgatggcata aatggctaa
421 caatggatc ttctggccca gacaaatggggcata tgatggcata tgtaaaggatc aacggcataa
481 taacagacac tatcaagatc tgatggggcata atataatgg aacacaagag tctgaatgt
541 catgtgtaaa tggttctgtc ttatctgtaa tgccatggatc accaaatgtt ggacaggcct
601 catacaatgtt ctccagaatg gaaaaggggaa aataatgttca atcgtcgaa atgaatgccc
661 ctaattatca ctatggggaa tgctccgtgtt atcctgttccatgttca acatgtgtgt
721 gcaggatggata tcggccatgc tcgaatggcata tgatgggtgtc ttccaaacgg aatctggaa
781 atcagatagg atacatatgc atggggatgtt ccggagacaa tccacggccctt atgataaaga
841 caggcgttg tggtccatgtc tgcttataatg gacaaatgg agtaaaaggaa ttttcattca
901 aataccggcaa tggttggggatc ataggagaa ctaaaatggatc tagtcaaga aacgggtttt
961 agatgttggatc ggatccgaaatc tgatggactt ggacacacaa taatcttca aaaaacaa
1021 atatcgtagg aataaatggatc tggtcaggatc atagcggggatc ttgttgcgttcatccagaac
1081 taacaggcgttggatgttggatc agaccttgc tgatgggtgtc actaatacggaa gggcgacccaa
1141 aagagaaacac aatctggactt ggccggaaatc gcatatctt tggttgcgtt aacagtgcata
1201 ctgtgggttgc tgcttggccca gacgggtgttgc
```

### polymerase [Influenza A virus (A/Ck/HK/96.1/02 (H5N1))]

LOCUS AAT39038 759 aa linear VRL 31-MAR-2008  
DEFINITION polymerase [Influenza A virus (A/Ck/HK/96.1/02 (H5N1))].

#### ORIGIN

```
1 merikelrldl msqsrtreil tkttvvhdmai ikkytsgrqe knpalrmkw mamkypitad
61 kriemiper neqqgtlwsk tndagsdrvm vsplavtwm rnpqatgtvh ypkvkytfe
121 kverlkhtgf gpvhfrnqvkh irrrvdinp hadlsakeaq dvimevvfpn evgartltse
181 sqltitkekk eelqdcckia lmvaymlare lvrkrflpv aggtssvyle vhlhqgtcw
241 eomytppgev rnndvdqslv iaarnivrra tvsadplasl lemchstqig girmvdilrq
301 npteeqavdi ckaamglris ssfsffgftf krtsgssvvk eeevltnq tlkirkvhegy
361 eeftfmvgrra tailrkattrr liqlivsrgd eqsllaeiiv amvsfqeem ikavrgdlnf
421 vnrangrlnp mhqlrrhfqk dakklfqnwq iepidnvmgm igilpdmtps temslrgrvr
481 skmgvdeyss tekvvvsidr flrvrdqrgn vllspeevse tqgtkeltit ysssmmwine
541 gpeeslvlny qwiirnwetv kiqwsqdptm lynkmegef pqlsvpkavrg qysgfvtrlf
601 qqmrdrvlgf dtvqikllp faaappeqsr mqfssltnv rgsqmrlir gnspvfnynk
661 ttkrltvlgk dagaltedpd egtagvesav lrgfligk dkrygpalsi selsnlakge
721 kanvligqgd vvvlvmkrkrd ssiltdsqta tkirmain
```

## Lampiran 3 Data uji coba untuk pengujian ketiga

### Data 1 :

#### crinkly4-like protein [Musa acuminata]

LOCUS AAR60009 894 aa linear PLN 30-JUL-2004  
DEFINITION crinkly4-like protein [Musa acuminata].

#### ORIGIN

```
1 mnsvrsgfli qaaflvmsg ssnmvgvsgl smssiavsys engpvfcgls sdgshlvacf
61 gadasiyyga plrlplglt agdfgvccg ldtgqpcycw snifvkmgvp qpmaegafys
121 eisagddhlc alrkpaidsr kgaslidcwg ymttashefd gsvaaitags sqvcgwgsl etqqlslngh
181 rtafcwgdet sssgvslsrl nrlfqsisag gfhvcgiln sqvcwgwsl etqqlslngh
241 dvkvmvpdmv svvaggrfha cgikslldhka vcwgfmqlqns mpppkdsrl eiaagdyftc
301 gvlaatslra vcwgtgpws ipmavspgc asnpccgggyy efshtswgnk vckpadsrvc
361 lpcsvgcpwg tpcstpcnltd sdhgcefnsc scasvcvcssf csqskesksk rllslqmpif
421 iaeivfaai vssvfvlvacl yarhrlqscr clesavtsr krtysfhken vkvrpdleel
481 kirraqmfty gelekatsgf seeslvkgks fscfvfkglk dgtvvavkra irvsdvkkns
541 kefhtedll slrnhhahnl llgcyceegge rllvyeefmah gslyrhlgk dlclrkrlwd
601 vrrvtiaavqa argiehlhy acppvhirdi kssnlidhee hnarrvdgfl sllgpvdssss
661 plsepagtl gyldpeyyrl hylttksdvy sfgvllleil sgrkaidmqf eegnivewav
721 plkiagdisa ildpvlpkbaa dleakkiaa maskcvrmrg kdrlsmdkvt tleralall
781 mgspcneqpi lptevlvgss rlhnkasqrs snrscsendt ddqiyeyrap switfpsvts
841 sqrrkssase gdaegknseg rslghggagd glrcleeeig paspqenlfl qhnff
```

//

#### zonadhesin-related protein [Musa balbisiana]

LOCUS BAG70984 1184 aa linear PLN 19-NOV-2008  
DEFINITION zonadhesin-related protein [Musa balbisiana].

#### ORIGIN

```
1 mrknepntse pgqsvsgidl tggpscets nstasnldhi ekgelqespq shcafeqpmf
61 lgqpvlgt anlhehlgs kdkllkkkrsp linkvcsre edlslkkqqqe itdfnmhkke
121 lelnlkkahd kdlleindlv mnsadkndki rllkeeflkk mtglekqmdc qykkllkdml
181 fardkeaqik nhweleakag kltessfridp lsdsgfrilee lklangdeah dgllrnrnyds
241 resgpfqnkq tgellitvadl vtnglnlkts egptvfppeg sgclpnqids lvsqsnvtyv
301 tetesepret plevpsllp stvdltmgt eseslpseap vmmsviieas slpsesresg
361 plqnktqgel itvdalvtng lncktsdgl ffppepgsgcl pnqidslvsq snvmytete
421 sepretplev pstlppsktv dltingteylp seapvmsli teaesgplqn kqtgeliiiva
481 dlvtsglnsk tseeltvfp egsgclpnqj nsilvsqsnvm yvtkttesdpn etplevpsl
541 ppstkvdlti tgeslpseap vmmsviiead dipinsstpa tvetekqrqa ensdilcssf
601 cplgsieegr smndgedasl ntallqnqan chtdfqndvs pcqaphfsrn eriensagie
661 vqsgsllqet plnlaftvg ltlnhnmvtvn sshgplpvct vpvlpvshvqe siaasgtee
721 lsdqiqhllsh qndmpmqepi vlsselvqqa npqhlvsqfv dhsflpnstd rsgrsisedl
781 rstsqpesvh yplfplaqlm ptqgiqepel kneltsirm qdkitkmhdd rkhlkyecd
841 gelekvrky dmllgdaese flrskevlet iynkvsrnq laefrakfi enkggtsssr
901 ghqtlqlqq ssqopqfijrs vssststmp lpatlppaaa psslasmvrlq tpmipssqtp
961 gssvrlqtpl ipyqgqapssl asvrlqappl pyqgqvpsla svrlqaplip sgqaarwtss
1021 vhpsnsarah lcpmvpqppn vqirsetrap aphlrqfrgn tsmtshqphq advlsatmgq
1081 vvpmpataqpc tdpfhfgqshi sntsvvcqda spfysepmg vgntiggvnr sisqftdlvp
1141 sfdswwlatkp tstatapfsq cvepcgvvdm ppsnsdvci sddd
```

//

Data 2 :

**cytochrome b [Phodopus campbelli]**

LOCUS ACG76157

380 aa

linear ROD 09-FEB-2009

DEFINITION cytochrome b [Phodopus campbelli].

ORIGIN

1 mtnmrknphl mkivnhafid lpapsnissw wnfsglglc lamqiltglf lamhytsdtt  
61 tafssvthic rdvnygwlr ylhangaasmf ficlfvhgr giyygsytmv etwnigvvll  
121 ltvmafamg yvlpwggmsf wgtavitnll saipyigttl vewiwiggsfv dkatltrffa  
181 fhfilpfift alvmvhllfl hetgsnppsg lnsdadkipf hpytikdlf gvfllvttm  
241 ilvlfpfdil gdpdnypnpan plntpahikp ewyflfayai lrsipnklgg vlalilisili  
301 lsilplhts khramtfrpl tqfmywlva dllyltwigg qpveypfii gqlasimyfa  
361 tivifmpias miednmlkfa

//

**exostosin 1 [Cricetulus griseus]**

LOCUS AAF71276

746 aa

linear ROD 21-MAY-2000

DEFINITION exostosin 1 [Cricetulus griseus].

ORIGIN

1 mqakkryfil lsagsclall fyfgygvfpra srshsrreheh sgrnglhqps pdhfwprrfad  
61 alhpffpwqdq lenedsgvhv sprqkrdrans svykgkkcrn escfdfalck kngkvyyvp  
121 qqkgekiae yqnillaaieg srfytsdpsq aclfvlsldt ldrdqqlspqy vhnrlsrkvqs  
181 lhlwnngrnh lifnllysgtwd pdytedvgfd iqgamlakas istenfrpnf dvslplfskd  
241 hprtgergf lkfn tipplr kymlvfkgkr yltgigsdtr nalyvhvngc dvllittckh  
301 gkdwqkhkds rcdrdn teye kydiyremln atfcvprgr rlgsfrflea lqaacvpml  
361 sngwelppfse viwnnqaavi gderllqip stirsihndt ilalrqqtgf lweayfssve  
421 kivlttlei qdrifkhisr nsliwnkhpg glfvlpqyss ylgdfpyya nlgkppskf  
481 tavihavtpl vsqsqpvlkt lvaaaksqyc aqivlwncd kplpkahrwp atavpvivie  
541 geskmrssrf lpydnniitda vlsldedtvl sttevdafat vwgsfperiv gyparshfwd  
601 nskerwgyts kwtdndsmvl tgaiyahkyy hlythyhpa slknmydqla nceditmnfl  
661 vsavtklppi kvtkkqkyke tmngqtsras rwadpdhfaq rqscmtfas wfgymplihs  
721 qmrldpvfk dqvsilrkky rdierl

//

### Data 3 :

#### NADH dehydrogenase subunit 5 [Panthera leo]

LOCUS AAM46197 606 aa linear MAM 02-JUN-2002  
DEFINITION NADH dehydrogenase subunit 5 [Panthera leo].

#### ORIGIN

```
1 mnlfasfmlt smfmllppi msntqlyknn lyphyvktti syafaismip ammfvssgqe
61 tivsnwhwls igtlklslsf kldyfsiif pvalfvtsi mefsmwyh dpyinrrfk
121 llmflitmmi lvtannlfal figwegvgim sflligwyyg radantaalq ailynrigdv
181 gfitamawfl amnawdfqq ifitqhknl ipllgllaa tgksaqfgh pwlpssamegp
241 tpvsallhss tmvavgvfl irfyplmeqn ktmqtltcl gaitlftai caltqndikk
301 ivafstssql gmlivtgvn qphlaflhic thaffkamlf mcsgsiihsl ndeqdirkmg
361 glykppmftt sssiigsl tgmpfltgfy skdliietan tsytnawall vtliatslta
421 aystrimffa llgqprfnsl spinenphl insikrllg sifagylish nippptipqm
481 tmyphylkta lattingfil alelnlaakn lkfkypsnif kfsnllyfp ivmrhpst
541 sltmsqksas mlldmiwles vlpksishfq mkmssttvsnq kglvklyfls fmitltsll
601 llsfhe
```

//

#### NADH dehydrogenase subunit 5 [Panthera pardus]

LOCUS YP\_001874850 606 aa linear MAM 14-APR-2009  
DEFINITION NADH dehydrogenase subunit 5 [Panthera pardus].

#### ORIGIN

```
1 mnlfiffmlt amfillppi msntqlyknn lyphyvktti syaftismip ammfvssgqe
61 tivsnwhwls igtlklslsf kldyfsiif pvalfvtsi mefsmwyh dpyinrrfk
121 llmflitmmi lvtannlfal figwegvgim sflligwyyg radantaalq ailynrigdv
181 gfitamawfl amnawdfqq ifitqhenln ipllgllaa tgksaqfgh pwlpssamegp
241 tpvsallhss tmvavgvfl irfyplmeqn ktmqtltcl gaitlftai caltqndikk
301 ivafstssql gmlivtgvn qphlaflhic thaffkamlf mcsgsiihsl ndeqdirkmg
361 glykppmftt sssiigsl tgmpfltgfy skdliietan tsytnawall vtliatslta
421 aystrimffa llgqprfnsl spinenphl insikrllg sifagylish nippptipqm
481 tmyphylkta lattingfil alelnlaakn lkfkypsnif kfsnllyfp ivmrhpst
541 sltmsqksas mlldmiwlen vlpksishfq mkmssttvsnq kglvklyfls fmitltsll
601 llsfhe
```

//

### Data 4 :

#### ZHX1 [Pongo pygmaeus]

LOCUS ABM89352 873 aa linear PRI 26-SEP-2008  
DEFINITION ZHX1 [Pongo pygmaeus].

#### ORIGIN

```
1 masrrkstt cmvlaseqdp dlelisldle gppvltpven traesissde evhesvdsdn
61 qqnkvvegg yckyctfqtp dlnmftfhvd sehpnvvlns syvcvcnfl tkrydalseh
121 nlkyhpgreen fkltmkvrnn qtifeqtind ltfdfgsfvke enaeqaeste vsssgisisk
181 tpimkmmknk venkriavhh nsvedvpeek eneikpdree tvenpssas esntstsvi
241 rihpstastv vtpaavlpql aqvitavsaq qnsnlipkvl ipvnsiptyn aaldnplll
301 ntynkfpypt mseitvlsaq akyteeqiki wfsaqrkhg vswtpheeve arrkqfnktv
361 htvpqtitvi pthistgsng lpsilqtcqf vgqpgvlvtq vagnttlpvt apialtvagv
421 psqnnvksa vpaaqptaet kpataavpts qsvkhetalv npdsfgirak ktkeqlaelk
481 vsylknqfph dseiqlmki tgltkgeikk wfsdtryngq nsksnqchl nndsstiiii
541 dssdettesp tvgtvqpkqs wnpfpdfptq kfekteaeql rvlqasflnn svltdeelnr
601 lrqatkktrr eidaftwkk kskalkeeke eidesnagss keeagetspg desgapksgs
661 tgkickkktpc qthmlksafv rtqwpspeey dklakesgl rtdivswfgd tryawkngnl
721 kwyyqysan sssmnglssl rkrgrrgpkgrgrprqrgp rgskrinnwrd rgpslikfk
781 gtaalkdyyl khkfneqd delvnkshmg yeqvrewfae qrerrsiegie lfeneeedee
841 viddqeedee etddsdstwep prhvkrklks sdd
```

//

**RecName: Full=Cystic fibrosis transmembrane conductance regulator; Short=CFTR;  
AltName: Full=Channel conductance-controlling ATPase; AltName: Full=cAMP-dependent  
chloride channel; AltName: Full=ATP-binding cassette transporter sub-family C member**

**7**

**LOCUS Q2IBE4 1480 aa linear PRI 07-JUL-2009  
DEFINITION RecName: Full=Cystic fibrosis transmembrane conductance regulator;  
Short=CFTR; AltName: Full=Channel conductance-controlling ATPase;  
AltName: Full=cAMP-dependent chloride channel; AltName:  
Full=ATP-binding cassette transporter sub-family C member 7.**

**ORIGIN**

1 mgrsplekas vvsdklfswt rpilkkgyrq rlelsdiyqi psadsadnl s eklerewrdre  
61 laskknpkli nalrrccffwr fmygqiflyl gevtkavql llgrriasyd pdnkeersia  
121 ylgigllcll fiavrllhp aigflhhqm qmrqamfsli ykktlklssr vldkisigql  
181 vsllsnlnk fdeglalahl wviapilqval lmgiwellq asafcgfqlf ivlalfqagl  
241 grmmmkkyrdq ragkinerly itsemieniq svkaycweea mekmienlrq telkltrkaa  
301 yvryfnssaf ffsgffvvfl svlpalyqkg ivlrkiftft sfcivlrmav trqfpawgt  
361 wydsgainke iqdfllqkgey ktleylnltt evmvenntaf weegfelfe kakqnnnre  
421 tsngddslf sfnfslgtpv lkdnfkier gqllavagst gagktsslmm imgelepseg  
481 kikhksprif csqfswwmpq tikenihfgv sydeyryrsv ikaclqeedi skfaekdniv  
541 lgegeitlg qqrarilar avykdadlyl ldsfpglydv letekefesc vcklmanktr  
601 ilvtksmehl kkadkilih egssyfygtf selqnlpdpf ssklmgcds dqfsaerrns  
661 itletelkq legdapwsrlt etkqkpfkqtf gefgekrkns ilpninsrk fsivqkplq  
721 mngeeedds pfeerrvslpv dseqealp ristvstpm lgarrqsvl nlmtgsvnqg  
781 qnighrktas trkvslapqa nlteldiysr rlsqetglei seeineedlk ecffadmesi  
841 pavitwplty ritvhksli fviwlcvlf laevaaslvw lwlgntplq dkgnsthsrn  
901 nsyaviitst ssyyvfyiyv gvadtlallm ffrglplvht litvskilhn kmlhsvlqap  
961 mstlnltag gilnrfskdi ailldlplt ifdfiqllli vigaiavvav lpqyifvatv  
1021 pivatvafq afqlqtsql kqlesegrsp ifthvlvtslk glvtrrafq qpyftrfk  
1081 alnlhtanwf lytlstlwfq mriemiflyi fiavtfisil ttgegegrvg iiltlannim  
1141 stlwavwns idvdslmrsv srvfkfidmp tegkptkstx pyknqqlskl miienshvkk  
1201 diwpsqgqm tvkdlktayt egnnaileni sfsispgorv gllgrrtgsgk stllsafirl  
1261 lnteqeiqid gwsdtsitlq qwrkfqvpv qkvfifsgtf rknlldpyeqw sdqeiwkvd  
1321 evglrsvieq fpkglfdflv dggcvlshgh kqlmclarsv lskakillld epsahldpvt  
1381 yqirrtlkq afactdvtlc hrieamlec qqfvlieenk vrgydsiqklnerslfqqa  
1441 ipsdvrklf phrnsskcks kpqiaalkkee teeeqvqdtl

//

## Data 5 :

**exostosin 1 [Cricetulus griseus]**

**LOCUS AAF71276 746 aa linear ROD 21-MAY-2000  
DEFINITION exostosin 1 [Cricetulus griseus].**

**ORIGIN**

1 mqakkryfil lsagsclall fyfggvqfra srshsrrehe sgrnglhqps pdhwprfad  
61 alhpffpwq lenedsgvhv sprqkrdans svykgkkcrn escfdfalck kngfkvyyp  
121 qqkgekiae yqnilaaieg srfytsdpsq acflvlsldt ldrdqslspay vhlrskvqs  
181 lhlwnngrnh llnflysgtw pdytedvgdf igqamlaks istenfrpnf dvsiplfskd  
241 hprtgergf lkftntipplr kydyremrlhn atfcflvprgr rlgssfrflea lqaacvpvml  
301 gkdwqkhds rcdrdnteye kydyremrlhn atfcflvprgr rlgssfrflea lqaacvpvml  
361 sngwelpnfse vijnwnqaaavt gderlllqip stirsihqdk ilalrqqtqf lweayfssve  
421 kivlttleii qdrifkhisr nsliwnkhpg glfvlpqyss ylgdfpyya nlglkppskf  
481 tavihavpl vsqsgpvlki lvaakaqsyc aqivlwncd kplpkahrwp atavpvivie  
541 geskvmsrfsf lpydnniitda vlsldedtvlt sttevdafat vwgsfperiv gyparsfhwd  
601 nskerwgys kwtdndsmvl tgaaiahkyy hlythylna slknmdqla ncidilmnf  
661 vsavtklppi kvtkkgyke tmqmgtsras rwadpdhfaq rqscmtnfas wfgymplhs  
721 qmrldpvlfk dqvsilrkky rdierl

//

**GPI-anchored ceruloplasmin [Rattus norvegicus]**

LOCUS AAF34175 1084 aa linear ROD 14-FEB-2000  
DEFINITION GPI-anchored ceruloplasmin [Rattus norvegicus].

**ORIGIN**

1 mkflllsall flhsslawtr ekhyigite awddyasgse ekelisvdtc qsnfyrlngp  
61 drigrkykka lyseytdgtf tktidkpawl gllgpvikae vgdkvsvhv kfsasrpptyfh  
121 ahgvttykan egaiydpntt dfqraddklf pgqqlvlylr anespseggd sncvtriyhs  
181 hvdapdkias glglplilck kgslhkekee nldqefvilmf svvdenlswy lednktfcy  
241 epekvdkne dfgesnrmys ingtftgslp glsmcaedrv kwylfgmgne vdvhselfhg  
301 qaltsknyht diinlpatl idvsmvaqnp gwmlscqnln nhlkaglgaf fqvrdcnkps  
361 pdddiqdryh rhyiaaeet iwdyapsgtd tftgentltsl gsdsrvffeq gatriggskyk  
421 klvyreytdt sftnrkqrgp deehlgilgp viwaevgdii rvtfhnkgqf plsiqpmgvyr  
481 ftkeneqyy gdgrsskqa shvapkettf yewtpkemg ptyadpvcls kmysgvdlt  
541 kdiftglipg mkickkgsnlk adgrqkdvdk efylfatvfd eneslllddn irmttopen  
601 vdkdedfqe snkmhsmnfg mygnlpqlnm clgesiwyl fsagneadvh giyfsgntyl  
661 skgerdtn lfphksstil mtpdtegsfd veclttdhyt ggmqkkytvn qckqgfedvt  
721 lgertyyi aavevewd psrdwemelh hlgeqnvsna fldkeeffig skykkvvrye  
781 ftdstfreqv krraeehng ilgplihadv gdkvkvaefn masrpyisla hgvktsstv  
841 tptlpgevrt yiwlpergs agtedspcip wayystdrv kdlysliipg livcrkslyvk  
901 vfpnkkkmey sllflvnden eswylddnin tsysdhpekvn kdneefiesn kmhaingkmf  
961 gnlqgltmhv gdevnwyma mgneidlhtv hfghqsfqyk hrgihssdf dlfpgtyqlf  
1021 emfpqptgtw llhchvtdhi hagmvtttyv lnpqassqsy rmtwnilytl lismttlfqi  
1081 stke

//  
UNIVERSITAS BRAWIJAYA



UNIVERSITAS BRAWIJAYA

